Event Replicator Administration

Using Event Replicator Administration

Version 2.6.1

June 2014

Adabas Event Replicator

This document applies to Event Replicator Administration Version 2.6.1.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: ARM-AARMUSER-261-20140626

Table of Contents

Preface

This document describes how to use Event Replicator Administration to maintain, activate, deactivate, open, and close replication definitions in the Replicator system files associated with each Event Replicator Server. You can also use Event Replicator Administration to review alter replication settings of Event Replicator Servers.

This document covers the following topics:

Selecting Event Replicator Servers	Describes how to select an Event Replicator Server for processing and review.
Maintaining Replication Definitions Using Event Replicator Administration	Describes how to maintain Event Replicator definitions using Event Replicator Administration.
Reviewing Active Replication Parameters of Event Replicator Servers	Describes how to review the replication parameters of Event Replicator Servers in Adabas Manager.
Reviewing the Replication Statistics of Event Replicator Servers	Describes how to review the replication statistics of Event Replicator Servers.
Managing File Replication of Adabas Files	Describes how to use Adabas Manager to manage file replication for those Adabas databases for which replication has been turned on.
Initiating a Replay Request Using Event Replicator Administration	Describes how you can initiate synchronized and replay-only replay processing using Event Replicator Administration and the ADARPL utility.
Reviewing and Managing the PLOG Data Set List	Describes how you can use Event Replicator Administration to review and maintain the list of PLOG data sets stored in the Replicator system file. It also describes the information stored for each PLOG data set.
Submitting Event Replicator Target Adapter Requests	Describes how you can populate, refresh (clear), and delete RDBMS tables using the Event Replicator Target Adapter once it is activated. You can submit requests to populate RDBMS tables with initial-state data, clear tables, or drop tables.

1 Selecting Event Replicator Servers

Database entries listed in Event Replicator Administration are preceded by a stoplight icon:

- Green indicates that the database is active.
- Yellow indicates that the database is locked.
- Red indicates that the database is inactive.

Only active Event Replicator Servers can be selected for further review or update in Event Replicator Administration.

To select an Event Replicator Server to work with:

1 Click and expand **Adabas Manager** *v.r.s* in tree-view, where *v.r.s* represents the version, release, and system maintenance level of the Adabas Manager installation.

2 Click and expand **Replicators** in tree-view.

3 Click on the database entry under **Replicators** in tree-view.

Once a database is selected, the following property information about that database appears in detail-view:

Property	Description
Database ID	The number of the database.
Database Name	The name of the database.
Database Version	The version level of the Adabas database software.
Database Load Date	The date and time the database was loaded.
Maximum Number of Files	The maximum number of files permitted for the database.
Number of Files Loaded	The total number of files currently loaded in the database.
Highest File Loaded	The highest file number currently in use in the database.

Property	Description
System File(s)	The number of Adabas system files allocated to the database.
Universal Encoding Support	Indicates whether Universal Encoding Support (UES) is active or inactive for the database.
Replication Server	Indicates whether or not the database is an Event Replicator Server.
Replication Server Version	Identifies the version number of Event Replicator.
Current Log Tape Number	The number of the most recent data protection log tape for the database
RABN Size	Indicates whether 3- or 4-byte RABNs are being used for the database

2 Setting Global Values

You can set the following global values in the Replicator system file using the Event Replicator Administration screens.

To set global values in the Replicator system file using the Event Replicator Administration:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click on **Replication Definitions** in tree-view under the selected database.

A table listing the global value settings in the Replicator system file appears in detail-view.

3 Click on the **Modify** button at the bottom of the detail-view.

The global value parameters you can modify become editable in detail-view.

4 Update the global parameters in detail-view as described in the following table.

Parameter Name	Specify	Default
Subtask (SUBTASKS parameter)	The number of subtasks in the Event Replicator Server.	0
Maximum Output Size (MAXOUTPUTSIZE parameter)	The maximum output message size for the Event Replicator for Adabas. The minimum value you can specify for this field is 32768. This parameter may be specified in bytes or it may be specified with the suffix K to indicate kilobytes. The maximum value is 2,147,483,647 bytes. The practical maximum is limited by the region size of the Event Replicator Server. One output buffer is acquired for each output task.	32768
EntireX Broker Stub Name (ETBBROKERNAME parameter)	The default webMethods EntireX Broker stub name to be used by the Event Replicator Server if no other name is specified.	BKIMBTSO

Parameter Name	Specify	Default
	webMethods EntireX Broker is a component of webMethods EntireX.	
Input Request Message Limit (IRMSGLIMIT parameter)	The maximum number of input request error messages issued by the Event Replicator Server during the interval set by the **Input Request Message Interval** field (also on this screen).	3
Input Request Message Interval (IRMSGINTERVAL parameter)	The interval during which the message limit specified by the **Input Request Message Limit** field applies.	1
Verify Mode (VERIFYMODE parameter)	Whether the Event Replicator for Adabas should run in verify (test) mode or not. Valid values are "Yes" (run in verify mode) or "No" (do not run in verify mode).	No
Format Buffer Validation (FBVALIDATION parameter)	Indicates the level of format buffer validation that should occur for subscriptions. Valid values are a blank, "Abort Reptor...", "Deactivate subscription and print warning", "no initial format buffer validation", or "Print warning message". ■ If a blank or "no initial format buffer validation" are specified, format buffer validation is not performed at Event Replicator initialization, at the initial handshake of databases, or when an updated FDT is received. Format buffer validation still occurs during the subscription phase of transaction processing, with validation errors written to the URBRRSP field of the Event Replicator URBR record. ■ If "Abort Reptor..." is specified, format buffer validation is performed. If validation errors are identified at Event Replicator initialization, the Event Replicator Server is terminated. If validation errors are identified after Event Replicator initialization, warning messages are issued for each format buffer in error. ■ If "Deactivate subscription and print warning" is specified, format buffer validation is performed. If validation errors are identified, the subscription for which format buffer validation failed is deactivated and warning messages are issued for each format buffer in error. ■ If "Print warning message" is specified, format buffer validation is performed. If validation errors are identified, warning messages are issued for each format buffer in error.	no initial format buffer validation
Record PLOG Information (RECORDPLOGINFO parameter)	Indicates whether or not PLOG information is saved in the Replicator system file. Valid values are "Yes" (store the information) or "No" (do not store the information). **Note:** If this parameter is set to "Yes", the PLOG information screens in Event Replicator Administration will either not display any PLOG data set information or will display outdated	No

Parameter Name	Specify	Default
	information on the screens. For more information about the PLOG information screens, read *Reviewing and Managing the PLOG Data Set List*, elsewhere in this guide.	
Retry Count (RETRYCOUNT parameter)	The default number of times that an attempt to open a destination or input queue will be attempted at the interval specified by the Retry Interval parameter. This is the equivalent of specifying the RETRYCOUNT parameter in the Event Replicator Server startup job. Valid values range from 0 through 2,147,483,647. A value of zero indicates that no retry processing should occur for any affected destinations.	10
Retry Interval (RETRYINTERVAL parameter)	The default number of seconds between retry attempts that will be performed for any destination or input queue for which no specific retry interval has been specified. This is the equivalent of specifying the RETRYINTERVAL parameter in the Event Replicator Server startup job. Valid values range from 0 through 2,147,483,647. A value of zero indicates that no retry processing should occur for any affected destinations.	0
Maximum Decompress Record Length (MAXRECORDSIZE parameter)	The maximum length (in bytes) of any decompressed record that can be processed by the Event Replicator Server. This is the equivalent of specifying the MAXRECORDSIZE parameter directly in the startup job. The minimum value you can specify for this field is 1. For an Event Replicator Server running with Adabas 7, the maximum value that can be specified for MAXRECORDSIZE is 32,767 bytes. However, in Adabas 8 systems, this limit has been lifted; the size of a decompressed record may be much larger than 32,767 bytes. Therefore, for an Event Replicator Server running with Adabas 8, the maximum value that can be specified for MAXRECORDSIZE is the larger of either 32,767 bytes or 50% of the setting of the ADARUN LRPL parameter.	32,767 bytes
Maximum Variable Record Length (MAXVARRECORDSIZE parameter)	The maximum length (in bytes) of variable decompressed records that can be processed by the Event Replicator Server. This is the equivalent of specifying the MAXVARRECORDSIZE parameter directly in the startup job. The value specified for MAXVARRECORDSIZE must be less than or equal to the setting of the MAXRECORDSIZE setting. The minimum value you can specify for this field is 1. For an Event Replicator Server running with Adabas 7, the maximum value that can be specified for MAXRECORDSIZE or MAXVARRECORDSIZE is 32,767 bytes. However, in Adabas 8 systems, this limit has been lifted; the size of a decompressed	32,767 bytes

Parameter Name	Specify	Default
	record may be much larger than 32,767 bytes. Therefore, for an Event Replicator Server running with Adabas 8, the maximum value that can be specified for MAXRECORDSIZE or MAXVARRECORDSIZE is the larger of either 32,767 bytes or 50% of the setting of the ADARUN LRPL parameter.	
Subtask Activation Wait Time(SUBTASKWAIT parameter)	The number of seconds that can be used to override the default time to wait for a subtask to finish initialization and activate. Valid values are from 1 to 3600 seconds. This is the equivalent of specifying the SUBTASKWAIT parameter in the Event Replicator Server startup job.	60
Log Input Transactions (LOGINPUTTRANSACTION parameter)	Whether or not the Event Replicator should use its SLOG system file as a temporary storage location for incoming compressed replication transactions, before they are queued for processing. Once transactions have been written to the SLOG system file, the Event Replicator Server processes them using a throttling mechanism so that only a limited amount of Event Replicator Server replication pool space is used at a time. Valid values are ALL (indicating that input transactions will always will be written to the SLOG system file), NO (the default, indicating that input transaction will not be written to the SLOG system file, or an integer in the range from 1 to 99. The integer setting specifies a threshold percentage of the LRPL (Event Replicator Server replication pool space) that can be used before triggering the writing of input transaction to the SLOG system file. This is the equivalent of specifying the LOGINPUTTRANSACTION directly in the Event Replicator Server startup job.	All
Global OPEN Value(GOPEN parameter)	Indicates whether or not destinations with **Open at Start** (DOPEN) set to "Global" in their definitions should be opened at Event Replicator Server startup. This parameter specifies the global policy for determining whether destinations with **Opened at Start** set to "Global" are opened at Event Replicator Server startup. Valid values are "Yes" or "No", with "Yes" as the default. When this parameter is set to "Yes", any destinations with **Opened at Start** set to "Global" in their definitions are opened at Event Replicator Server startup. When this parameter is set to "No", any destinations with **Opened at Start** set to "Global" in their definitions are *not* opened at Event Replicator Server startup. This is the equivalent of specifying the GOPEN directly in the Event Replicator Server startup job.	Yes

Parameter Name	Specify	Default
Maximum RPL Usage (TLMAX parameter)	The maximum percentage of the Event Replicator replication pool that can be used for transaction log (TLOG) processing. Valid values range from "0" through "100".	40
Restart RPL Usage (TLRESTART parameter)	The amount of available Event Replicator replication pool storage that must be available before transaction logging (TLOG logging) can restart. Valid values range from "0" through "99".	50
Input Queue Level (TLINPUT parameter)	The level of transaction logging that should occur when a transaction is taken off the input queue and put on the transaction assignment queue. Valid values are "no logging", "log event and input transaction data", "log event, input transaction, and file/ record data", or "log event and all available input information for the event".	no logging
No Match Level (TLNOSUB parameter)	The level of transaction logging that should occur when a transaction is not queued to any subscription in Event Replicator. Valid values are "no logging", "log event and input transaction data", "log event, input transaction, and file/record data", or "log event and all available input information for the event".	no logging
Queue Completion Level (TLQCOMP parameter)	The level of transaction logging that should occur prior to a transaction being assigned to the completion queue. Valid values are "no logging", "log event and input transaction data", "log event, input transaction, and file/record data", or "log event and all available input information for the event".	no logging
Completion Level (TLCOMP parameter)	The level of transaction logging that should occur when a transaction has been fully processed and Adabas has been informed that the transaction was successfully replicated. Valid values are "no logging", "log event and input transaction data", or "log event, input transaction, and file/record data".	no logging
Request Received Level (TLREQRECV parameter)	The level of transaction logging that should occur when a user request has been received. Valid values are "no logging", "log event but no data", or "log event and the entire input buffer before and after translation if appropriate".	no logging
Request Rejected Level (TLREQREJECT parameter)	The level of transaction logging that should occur when a user request is rejected due to an error in interpreting the request. Valid values are "no logging", "log event but no data", or "log event, error code, and entire input buffer".	no logging
Request Error Level (TLREQERR parameter)	The level of transaction logging that should occur when a user request is rejected due to an error in carrying out the request. Valid values are "no logging", "log event and URBS", "log event and URBI", or "log event, URBS, and URBI".	no logging
Status Request Level (TLSTATUS parameter)	The level of transaction logging that should occur when a user request for status on an Event Replicator resource has been processed. Valid values are "no logging", "log event and URBS", "log event and URBI", or "log event, URBS, and URBI".	no logging

Parameter Name	Specify	Default
I-State Start Request Level (TLISTATE parameter)	The level of transaction logging that should occur when a user request for initial-state information for a file has started. Valid values are "no logging", "log event and URBS", "log event and URBI", or "log event, URBS, and URBI".	no logging
I-State Completion Level (TLISTATECOMP parameter)	The level of transaction logging that should occur when an initial-state information request has completed. Valid values are "no logging", "log event and URBS", "log event and URBI, if available", or "log event, URBS, and URBI, if available".	no logging
Retransmit Request Level (TLRETRANS parameter)	The level of transaction logging that should occur when a user request to retransmit a specific transaction has been processed. Valid values are "no logging", "log event and URBS", "v", or "log event, URBS, and URBI".	no logging

5 When all specifications have been made to your satisfaction, click **OK** to save the global values or click **Cancel** to cancel your updates..

3 Controlling Connections to Adabas Databases

By default, Event Replicator for Adabas attempts to connect with any Adabas database encountered during an Event Replicator Server session. You may, however, prefer to control these connection attempts using an Adabas database connection definition. This chapter describes how to maintain these definitions using Event Replicator Administration.

Listing Adabas Database Connection Definitions

To list the connection definitions for Adabas databases in Event Replicator Administration:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Replication Definitions** in the tree-view under **Replicators**.

A list of the Event Replicator definitions you can maintain in the Replication system file appears in tree-view.

3 Click on **Database IDs** in the tree-view under **Replication Definitions**.

A table listing the database IDs for which connection definitions have been defined appears in detail-view. This table lists the database IDs and whether or not they are connected to this Event Replicator.

Adding a Connection Setting for an Adabas Database

To add a connection definition for an Adabas database:

1 List the connection definitions in Event Replicator Administration, as described in *Listing Adabas Database Connection Definitions*, elsewhere in this guide.

The destinations are listed in the Database IDs table in detail-view.

2 Right-click on **Database IDs** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3 Click on **Create New Database ID** in the drop-down menu.

A blank **Database ID** panel appears in detail-view.

4 Fill in the fields in Value columns of the detail-view table as described below:

Parameter Name	Description	Default
Database ID (DATABASE ID)	Specify the database ID of the Adabas database to which this connection definition applies.	---
Connect at Replicator Startup(DBCONNECT)	Specify whether or not the Event Replicator Server should initiate an attempt to connect to the Adabas database at Event Replicator Server startup. Valid values are "Yes" or "No". If you specify "Yes", the Event Replicator Server will initiate an attempt to connect to the Adabas database after the Event Replicator Server starts. If you specify "No", the Event Replicator Server will *not* initiate an attempt to connect to the Adabas database. The database may connect with the Event Replicator Server at some point, but the Event Replicator Server will not initiate the connection at Event Replicator Server startup.	Yes

5 Click **OK** key to save the connection definition and return to the Database IDs screen.

The database connection definition is added.

Modifying the Connection Definition for an Adabas Database

To modify a connection definition in the Event Replicator Administration:

1 List the connection definitions in the Event Replicator Administration, as described in *Listing Adabas Database Connection Definitions*, elsewhere in this guide.

The destinations are listed on the Database IDs screen.

2 Locate the definition you want to modify in detail-view and click on the database ID.

The **Database ID** panel appears in detail-view showing the definition.

3 Click **Modify**.

The **Database ID** panel refreshes, allowing you to change the connection information for this database. For information on modifying this screen, read the description of *Adding a Connection Setting for an Adabas Database*, elsewhere in this section.

4 When all modifications have been made, click **OK** to save the changes and return to the Database IDs screen.

Copying the Connection Definition for an Adabas Database

To copy a connection definition in the Event Replicator Administration:

1 List the connection definitions in the Event Replicator Administration, as described in *Listing Adabas Database Connection Definitions*, elsewhere in this guide.

The destinations are listed on the Database IDs screen.

2 Locate the definition you want to copy in detail-view and click on the database ID .

The **Database ID** panel appears in detail-view showing the definition.

3 Click **Copy**.

The **Database ID** panel refreshes, allowing you to specify a new Adabas database ID for the new definition.

4 Specify a new Adabas database ID for the copy of the definition and click **OK**.

The connection definition is copied and the copy appears on the Database IDs screen.

Activating and Deactivating Databases and Files

You can use Event Replicator Administration to activate and deactivate databases or individual files within a database.

Caution: Be careful when you activate and deactivate replication definitions and databases, especially if replication is ongoing at the time. Whenever you activate or deactivate definitions or databases, you run the risk of altering what data is replicated and how that replication occurs. If the Event Replicator Server receives data from an Adabas database for which it has no active definitions, replication simply does not occur.

This section covers the following topics:

- Activating Databases
- Activating Files
- Deactivating Databases

- Deactivating Files

Activating Databases

To use Event Replicator Administration to activate a database:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Database IDs** in the tree-view under **Replication**.

A table listing the database connections in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the database you want to activate.

Details about the database appear in detail-view.

6 To activate the entire database (including all files in the database), click on the **Activate** button in detail-view.

The database and all of its files are activated.

Activating Files

To use Event Replicator Administration to activate an individual file:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Database IDs** in the tree-view under **Replication**.

A table listing the database connections in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the database containing the file you want to activate.

Details about the database appear in detail-view.

6 Click on the **Number of Files** field name in detail-view.

A list of the files in the database appears in detail-view

7 Click the **Activate** button next to the file (or files) you want to activate.

The file is activated.

Deactivating Databases

To use Event Replicator Administration to deactivate a database:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Database IDs** in the tree-view under **Replication**.

A table listing the database connections in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the database you want to deactivate.

Details about the database appear in detail-view.

6 To deactivate the entire database (including all files in the database), click on the **Deactivate** button in detail-view.

The database and all of its files are deactivated.

Deactivating Files

To use Event Replicator Administration to deactivate an individual file:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Database IDs** in the tree-view under **Replication**.

A table listing the database connections in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the database containing the file you want to deactivate.

Details about the database appear in detail-view.

6 Click on the **Number of Files** field name in detail-view.

A list of the files in the database appears in detail-view

7 Click the **Deactivate** button next to the file (or files) you want to deactivate.

The file is deactivated.

Deleting the Connection Definition for an Adabas Database

To delete a connection definition in the Event Replicator Administration:

1 List the connection definitions in the Event Replicator Administration, as described in *Listing Adabas Database Connection Definitions*, elsewhere in this guide.

The destinations are listed on the Database IDs screen.

2 Locate the definition you want to delete in detail-view and click on the database ID .

The **Database ID** panel appears in detail-view showing the definition.

3 Click **Delete**.

A verification panel appears in detail-view

4 Click **OK** to delete the definition or click **No** to stop the deletion.

The definition is deleted if you clicked **OK**; otherwise it is not.

4 Refreshing Definitions

If your Event Replicator Server definitions are stored in a Replicator system file, you can refresh them on the Event Replicator Administration screens. You can:

- Refresh all of the definitions in the Replicator system file at one time
- Refresh an individual definition in the Replicator system file.

Using the Event Replicator Administration screens, you cannot refresh all of the definitions of a specific type at one time. For example, while you can refresh a single subscription definition, you cannot refresh all subscription definitions at once using the Event Replicator Administration screens. To do this you would need to refresh all of the definitions in the Replicator system file -- regardless of definition type.

This is the same as running an RPLREFRESH command, except that you cannot refresh all of the definitions for a specific resource type. For more information about the RPLREFRESH command, read about the RPLREFRESH command in *Event Replicator for Adabas Administration and Operations Guide* provided with your Event Replicator Administration documentation.

Refreshing All Definitions

To refresh all of the definitions on the Event Replicator Administration screens:

1 Select an Event Replicator Server in tree-view as described in ***Selecting Event Replicator Databases***, elsewhere in this guide.

2 Click and expand **Replication Definitions** in tree-view under the selected database.

A subset of the replication parameters and definitions appear in detail-view.

3 At the bottom of the detail-view display, click the **RPLREFRESH All** button.

All replication definitions stored in the Replicator system file are refreshed on the Event Replicator Administration screens.

Refreshing Individual Definitions

To refresh an individual definition on the Event Replicator Administration screens:

1 Select an Event Replicator Server in tree-view as described in ***Selecting Event Replicator Databases***, elsewhere in this guide.

2 Click and expand **Replication Definitions** in tree-view under the selected database.

Or:

Click and expand **Active Parameters** and then click and expand **Replication** in tree-view under the selected database.

3 Click and expand the type of resource definition you are interested in refreshing. For example, if you are interested in refreshing an individual subscription definition, click and expand the **Subscriptions** list.

A list of the definitions of the type you selected appears in detail-view.

4 Locate the definition you want to refresh in the table in detail-view and click on it.

The definition appears in detail-view listing the current settings for the definition you selected.

5 At the bottom of the detail-view display, click the **RPLREFRESH** button.

A confirmation panel appears verifying that you want to refresh the definition. If you click **Yes** (indicating that you do want to refresh the resource), the replication definition stored in the Replicator system file is used to refresh the Event Replicator Administration screen. If you click **No** (indicating that you do not want to refresh the resource), the replication definition is not refreshed.

5 Maintaining Initial-State Definitions

An initial-state definition defines an initial-state request for data from the target application. Initial-state definitions identify the subscription, destination, and specific Adabas files to use in an Event Replicator for Adabas run; at least one subscription, destination, or file must be specified.

Initial-state data can contain any subset of the data on the Adabas database, based on the specifications in the initial-state definition and parameters supplied in the initial-state request. Records can be selected for initial-state processing in one of the following manners:

- The complete file can be selected.
- Records are selected from the file based on an ISN list.
- Records are selected from the file based on specified selection criteria.

Note: Each replicated initial-state record contains the related data storage after image. No before image is replicated for an initial-state record.

In addition, you can populate a database with initial-state data using the Adabas Event Replicator Subsystem or from a client application. For more information on populating a database with initial-state data from the Adabas Event Replicator Subsystem, read *Populating a Database With Initial-State Data*, in *Adabas Event Replicator Subsystem User's Guide*. For more information on populating a database with initial-state data from a client application, read *Event Replicator Client Requests*, in *Event Replicator for Adabas Programmer's Reference Guide*.

Listing Initial-State Definitions

To use Event Replicator Administration to list the initial-state definitions stored in the Replicator system file:

1 Select an Event Replicator Server in tree-view as described in ***Selecting Event Replicator Databases***, elsewhere in this guide.

2 Click and expand **Replication Definitions** in tree-view under the selected database.

3 Click on **Initial States** in the tree-view under **Replication Definitions**.

A table listing the initial-state definitions in the Replicator system file appears in detail-view.

Adding Initial-State Definitions

This section describes the steps you must complete to use Event Replicator Administration to add an initial-state definition to the Replicator system file:

Step 1. Access the Initial-State Definition Area of Event Replicator Administration

To access the initial-state definition area of Event Replicator Administration:

1 List the initial-state definitions in Event Replicator Administration, as described in *Listing Initial-State Definitions*, elsewhere in this guide.

The initial-state definitions are listed in detail-view.

2 Right-click on **Initial States** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3 Click on **Create New Initial-State** in the drop-down menu.

A blank **Initial State** panel appears in detail-view.

Step 2. Supply a Name and the Number of Concurrent Initial-State Requests Possible for the Initial-State Definition

To supply a name and the number of concurrent initial-state requests possible for the initial-state definition:

1 In the table on the blank **Initial State** panel, use the **Value** columns to supply values for the following parameters:

Parameter Name	Description	Default Value
Initial State Name (NAME)	Specify a unique name for the initial-state definition. The name must use alphanumeric characters and be between one and 8 characters long.	---
Number of Concurrent Initial State Requests (IMAXREQ)	Specify the number of concurrent initial-state requests that can occur in a given instance of this initial-state definition. Valid values range from 1 to 2,147,483,647. This is the same	1

Parameter Name	Description	Default Value
	as specifying the IMAXREQ parameter directly in the Event Replicator Server startup job.	

Note: The **Initial State Destinations**, **Initial State Subscriptions**, and **Initial State Files** parameters are not available immediately. These must be set by modifying the initial-state definition after it is created.

2 Click **OK** to add the initial-state definition to the Replicator system file.

Step 3. Select Destinations, Subscriptions, and Files for the Initial-State Definition

At least one subscription, destination, or file must be specified for an initial-state definition; all three may be specified, but at least one *must* be.

When one or more DBID/file combinations are specified in an initial-state definition, and one or more destinations or subscriptions are also defined, the initial-state data for the specified DBID/files are sent only to the specified destinations or as described by the specified subscriptions. In other words, delivery of the initial-state data in the files is restricted by the destination and subscription definitions. If no DBID/file combinations are specified in the initial-state definition, the specified destination and subscription definitions are used to construct a list of related DBID/files that should be used for the run.

This step describes how to:

- Select Destinations for the Initial-State Definition
- Select Subscriptions for the Initial-State Definition
- Select Files for the Initial-State Definition

Select Destinations for the Initial-State Definition

To select destination definitions for the initial-state definition:

The destination definitions must be previously defined.

1 Click **Modify** to modify the initial-state definition.

The **Initial State** panel refreshes, allowing you to specify destinations, subscriptions, and files for the initial-state definition.

2 Click on the check mark in the **Value** column for the **Initial State Destinations** parameter (IDESTINATION parameter) .

The **Destination List** selection panel appears and allows you to select (and remove) destinations for the replicated data that will be generated by this initial-state request. The destination definitions must be previously defined.

3 Select and remove destination definitions in the table on the **Destination List** panel:

- To select a destination for the initial-state definition, click on the name of a destination in the **Destinations Available** table and then click the right arrow button. The selected destination appears in the **Destinations Selected** table.
- To remove a destination from the initial-state definition, click on the name of the destination in the **Destinations Selected** table and then click the left arrow button. The selected destination appears in the **Destinations Available** table.

4 When the destinations appear correctly in the two tables of the **Destination List** selection panel, click **OK** to approve the selections. The **Initial State** panel appears again in detail-view.

Select Subscriptions for the Initial-State Definition

To select subscription definitions for the initial-state definition:

The subscription definitions must be previously defined.

1 Click **Modify** to modify the initial-state definition.

The **Initial State** panel refreshes, allowing you to specify destinations, subscriptions, and files for the initial-state definition.

2 Click on the check mark in the Value column for the **Initial State Subscriptions** parameter (`ISUBSCRIPTION` parameter) .

The **Subscription List** selection panel appears and allows you to select (and remove) subscriptions for the replicated data that will be generated by this initial-state request. The subscription definition must be previously defined.

3 Select and remove subscription definitions in the table on the **Subscription List** panel:

- To select a subscription for the initial-state definition, click on the name of a subscription in the **Subscriptions Available** table and then click the right arrow button. The selected subscription appears in the **Subscriptions Selected** table.
- To remove a subscription from the initial-state definition, click on the name of the subscription in the **Subscriptions Selected** table and then click the left arrow button. The selected subscription appears in the **Subscriptions Available** table.

4 When the subscriptions appear correctly in the two tables of the **Subscription List** selection panel, click **OK** to approve the selections. The **Initial State** panel appears again in detail-view.

Select Files for the Initial-State Definition

To select files for the initial-state definition:

1 Click **Modify** to modify the initial-state definition.

The **Initial State** panel refreshes, allowing you to specify destinations, subscriptions, and files for the initial-state definition.

2 Click on the check mark in the Value column for the **Initial State Files** parameter (`IDBID`, `IFILE`, `ISNLIST`, and `SELCRIT` parameters) .

The **IFILE List** selection panel appears and allows you to select (and remove) database files for the replicated data that will be generated by this initial-state request.

3 Select and remove database files on the **IFILE List** panel.

To select a database file for the initial-state definition, click on the **New** button. A blank line appears. Supply the following information for the database file

- In the **IDBID** column, specify the database ID associated with an input file you want used for this initial-state definition. The database ID is numeric and can range from one to 65535. There is no default.
- In the **IFile** column, specify the file number of an input file you want used for this initial-state definition. There is no default.
- In the **Filter Method** column, specify what filter method will be used to process data from this input file. Valid values are **All** (no filter method is used; all records from the input file are processed and replicated), **ISN List** (the only records processed are those with the ISNs you specify in the initial-state request), and **Selection Criteria** (the only records processed are those that meet the search criteria you specify in the **Selection Criteria** column). If you elect to use an ISN list, be sure to supply an ISN list when you submit the initial-state request. The ISN list can be specified using the Adabas Event Replicator Subsystem (read *Populating a Database With Initial-State Data* , in *Adabas Event Replicator Subsystem User's Guide*) or in the client request (read *ISN List Format* in the section entitled *Initial-State Requests* in *Event Replicator for Adabas Programmer's Reference Guide*).
- In the **Selection Criteria** column, specify any Adabas search criteria you want to use to select input data for processing (if **Filter Method** was set to **Selection Criteria**). If you specify search criteria, only the records in the input file that satisfy the search criteria will be processed. A maximum of a 60-byte search buffer is provided. There is no default.

To delete a database file from the list, click the check box in the corresponding **All** column to select it and then click the **Delete** button.

4 When the database files appear correctly in the table of the **IFILE List** selection panel, click **OK** to approve the selections. The **Initial State** panel appears again in detail-view.

Step 4. Save the Initial-State Definition

When all specifications have been made to your satisfaction, click **OK** to save the initial-state definition.

Modifying Initial-State Definitions

To use Event Replicator Administration to modify an initial-state definition in the Replicator system file:

1 List the initial-state definitions in Event Replicator Administration, as described in *Listing Initial-State Definitions*, elsewhere in this guide.

The initial-state definitions are listed in detail-view.

2 Locate the definition you want to modify in the table in detail-view and click on it.

The **Initial State** panel appears in detail-view listing the current settings for the initial-state definition you selected.

3 Click the **Modify** button.

The initial-state parameters you can modify become editable in detail-view. For information on modifying these parameters, read the description of *Adding Initial-State Definitions*, elsewhere in this section.

4 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the changes.

Copying Initial-State Definitions

To use Event Replicator Administration to copy an initial-state definition in the Replicator system file:

1 List the initial-state definitions in Event Replicator Administration, as described in *Listing Initial-State Definitions*, elsewhere in this guide.

The initial-state definitions are listed in detail-view.

2 Locate the definition you want to copy in the table in detail-view and click on it.

The **Initial State** panel appears in detail-view listing the current settings for the initial-state definition you selected.

3 Click the **Copy** button.

A copy of the initial-state definition is created and its parameter values appear in detail-view.

4 Specify a new, unique name for the copy of the initial-state definition in the Value column for the **Initial-State Name** parameter.

5 If you wish, modify any other parameters for the new copy in detail-view. For information on modifying the parameters, read the description of ***Adding Initial-State Definitions***, elsewhere in this section.

6 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the copy.

Deleting Initial-State Definitions

To use Event Replicator Administration to delete an initial-state definition in the Replicator system file:

1 List the initial-state definitions in Event Replicator Administration, as described in ***Listing Initial-State Definitions***, elsewhere in this guide.

The initial-state definitions are listed in detail-view.

2 Locate the definition you want to delete in the table in detail-view and click on it.

The **Initial State** panel appears in detail-view listing the current settings for the initial-state definition you selected.

3 Click the **Delete** button.

A confirmation panel appears verifying that you want to delete the definition. If you click **Yes** (indicating that you do want to delete the definition), the definition is deleted. If you click **No** (indicating that you do not want to delete the definition), the definition is not deleted.

6 Maintaining Destination Definitions

A destination definition defines the destination of the replicated data. Destination definitions can be created for Adabas, webMethods EntireX, WebSphere MQ, File, and Null destinations. At least one definition is required for every Event Replicator for Adabas destination you intend to use.

Destination Type	Description
Adabas	Data is replicated to one or more Adabas files.
webMethods EntireX	Replicated data is written to an output queue via webMethods EntireX.
WebSphere MQ	Replicated data is written to an output queue via IBM WebSphere MQ.
Null	Data replication is tested without actually sending the data to a destination.
File	Replicated data is written to the CLOG, using TLOG URBLTDOD records.

Listing Destination Definitions

To use Event Replicator Administration to list the destination definitions stored in the Replicator system file:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Replication Definitions** in tree-view under the selected database.

3 Click on **Destinations** in the tree-view under **Replication Definitions**.

A table listing the destination definitions in the Replicator system file appears in detail-view.

Creating an Adabas Destination Definition

Using Adabas destination definitions, data can be replicated to one or more Adabas files. This section describes how to create an Adabas destination definition using Event Replicator Administration.

Note: An Adabas destination can be referenced by no more than one subscription.

To create an Adabas destination definition in Event Replicator Administration, complete the following steps:

- Step 1. Access the Adabas Destination Definition Creation Area
- Step 2. Specify General and TLOG Adabas Destination Parameters
- Step 3. Specify Input and Target Adabas Destination Databases and Files
- Step 4. (Optional) Specify File-Related Parameters for the Adabas Destination

Step 1. Access the Adabas Destination Definition Creation Area

To access the Adabas destination definition creation area of Event Replicator Administration:

1 List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*, elsewhere in this guide.

The destination definitions are listed in detail-view.

2 Right-click on **Destinations** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

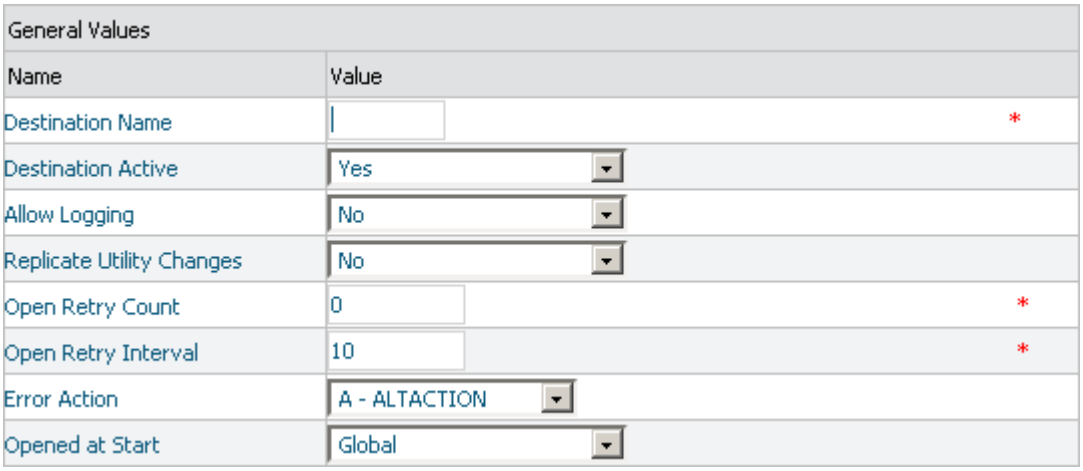
3 Click on **Create New Adabas Destination** in the drop-down menu.

A blank **Adabas Destination** panel appears in detail-view.

Step 2. Specify General and TLOG Adabas Destination Parameters

To use Event Replicator Administration to supply general and TLOG specifications for an Adabas destination definition, complete the following steps:

1 Supply general parameter specifications for the Adabas destination. The fields for these general parameters can be found at the top of the **Adabas Destination** panel:



Supply values for the these fields, as described in the following table:

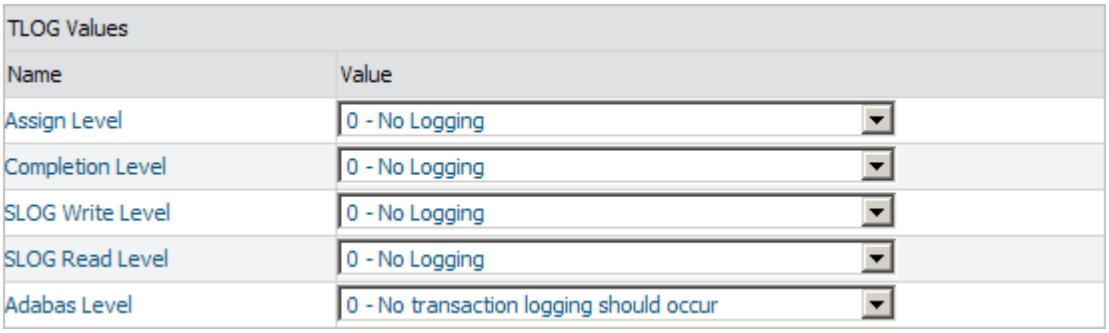
Parameter Name	Specify	Default
Destination Name (DESTINATION NAME)	The unique name for the Adabas destination definition. The specified name must be alphanumeric and be between one and eight characters long.	---
Destination Active (DACTIVE)	Whether or not this destination definition should be activated for use once it is loaded by the Event Replicator Server. Valid values are "Yes" (load and activate the definition) or "No" (load, but do not activate the definition).	Yes
Allow Logging (DLOG)	Whether or not subscription logging should be activated for this destination definition. Valid values are "Yes" (activate subscription logging) or "No" (do not activate subscription logging).	No
Replicate Utility Changes (DREPLICATEUTI)	Whether Adabas utility change replication should be activated for a destination at Event Replicator Server startup or whether it should be activated for a specific target file. This parameter appears twice on this screen. In the general location, it specifies whether the utility change replication should be activated for a destination at Event Replicator Server startup; in the location (next to the Target File parameter), it specifies whether the utility change replication should be activated for the associated target file. Valid values are "Yes" and "No". If "Yes" is specified, utility replication is activated; if "No" is specified, utility replication is not activated. For more information about replicating utility functions, read *Replicating Utility Functions*, in *Event Replicator for Adabas Concepts*.	No
Open Retry Count (DRETRYCOUNT)	The number of times that an attempt to open the destination will be retried at the interval specified by the **Retry Interval parameter**. This is the equivalent of specifying the DRETRYCOUNT parameter directly in the Event Replicator Server startup job. Valid values range from 0 through 2,147,483,647 or the literal "GLOBAL". If the value "GLOBAL" is specified for this parameter, the specification for the **Retry Count global variable** will be used. Any retry attempts will occur at the interval specified by the **Retry Interval** parameter. A value of zero indicates that no retry attempt to open this destination should occur.	The value of the **Retry Count global variable**.
Open Retry Interval (DRETRYINTERVAL)	The default number of seconds between retry attempts to open the destination. This is the equivalent of specifying the DRETRYINTERVAL parameter directly in the Event Replicator Server startup job. Valid values are 0, 5 through 2,147,483,647, or the literal "GLOBAL". If the value "GLOBAL" is specified for this parameter, the specification for the **Retry Interval global variable** will be used. A	The value of the **Retry Interval global variable**.

Parameter Name	Specify	Default
	value of zero indicates that no retry attempt to open this destination should occur. Except for a specification of zero, the minimum value that can be specified for this parameter is 5 seconds.	

Error Action (DAERROR)	The action to be taken when an error occurs during replication to an Adabas destination. This is the equivalent of specifying the DAERROR parameter directly in the Event Replicator Server startup job. Valid values are ALTACTION, BACKOUT, or CLOSE.
	
	In all cases (ALTACTION, BACKOUT, and CLOSE), if response 148 is returned and the SLOG system file is available, the destination is closed.
	
	If an insert, update, or delete operation fails because a replicated record already exists or does not exist, an appropriate message is issued. If the transaction fails because of an error, a message containing the two-character Adabas command, the database ID, the file number, the response code and the subcode is written. If the DATMETHOD is set to ISN, the text "ISN" will be appended to this message as well as the ISN value. If the DATMETHOD is set to KEY, the record key will be written in both hexadecimal and readable format in a separate message.
	
	Additional actions are taken, based on the value of this parameter. These actions are:
	
	■ A - ALTACTION: Processing continues with the next update that is part of the same transaction. Some special processing occurs when DAERROR=ALTACTION:
	  ■ If an insert is processed and the record already exists, the record is updated.
	  ■ If an update is processed and the record does not exist, the record is inserted.
	  ■ If a delete is processed and the record does not exist, processing continues with the next record.
	  For other errors, the record is skipped.
	■ B - BACKOUT: A message is issued indicating that the transaction will be backed out and then ignored. The current transaction is backed out and processing continues with the next transaction to be replicated.
	■ C - CLOSE: A message is issued indicating that the transaction will be backed out and the destination will be closed. The current transaction is backed out and transaction logging (to the SLOG file) will begin, if defined for the destination.
		A-ALTACTION

Parameter Name	Specify	Default
Opened at Start (DOPEN parameter)	Whether or not the destination should be opened at Event Replicator Server startup. Valid values are "Yes", "No", or "Global", with "Global" as the default. When this parameter is set to "Yes", the destination is opened at Event Replicator Server startup. When this parameter is set to "No", the destination is *not* opened at Event Replicator Server startup. When this parameter is set to "Global", the decision to open the destination at Event Replicator Server startup depends on the setting of the **Global OPEN Value** (GOPEN) global parameter. If GOPEN=YES, the destination is opened at Event Replicator Server startup; if GOPEN=NO, it is not opened. This is the equivalent of specifying the DOPEN parameter in the Event Replicator Server startup job.	Global

2 Supply TLOG parameter specifications for the Adabas destination. The fields for these TLOG parameters can be found at the bottom of the **Adabas Destination** panel:



Supply values for the these fields, as described in the following table:

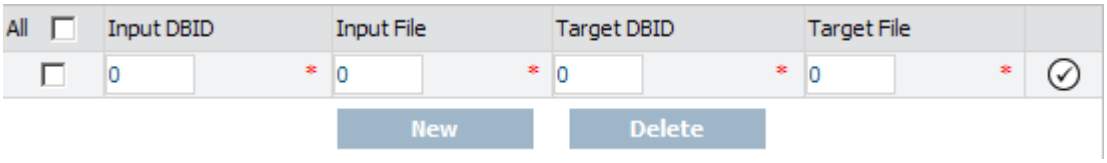
Parameter Name	Specify	Default
Assign Level (DTLASSIGN)	The level of transaction logging that should occur when a transaction is assigned to a destination for output processing. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Completion Level (DTLCOMP)	The level of transaction logging that should occur when a transaction has been successfully output to the messaging system. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Write Level (DTLSLOGWRITE)	The level of transaction logging that should occur when a transaction has been successfully written to the SLOG file. Valid values are "no	no logging

Parameter Name	Specify	Default
	logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	
SLOG Read Level (DTLSLOGREAD)	The level of transaction logging that should occur when a transaction has been successfully read from the SLOG and is about to be queued for output to the destination. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Adabas Level (DTLADABAS)	The level of transaction logging that should occur when a transaction for an Adabas destination incurred an error. This is the equivalent of specifying the DTLADABAS parameter in the Event Replicator Server startup job. Valid values range from 0 through 3, as described below: ■ 0: No transaction logging should occur. ■ 1: Log event and Adabas error information. ■ 2: Log event, Adabas error information, and file and record data. ■ 3: Log event, Adabas error information, file and record data, and the actual payload.	0

Step 3. Specify Input and Target Adabas Destination Databases and Files

To use Event Replicator Administration to supply input-target file pairings for an Adabas destination definition, complete the following steps:

1 The fields for input-target file pairing specifications in an Adabas destination definition can be found in the middle of the **Adabas Destination** panel:



For the first input-target destination file pairing, fill in the fields in Value columns as described in the following table:

Parameter Name	Specify	Default
Input DBID (DAIDBID)	The database ID associated with an input file (see the Input File field) for this Adabas destination. The database ID is numeric and can range from one to 65535. The input database ID and file specified must also be included in an SFILE definition in the subscription. Multiple input database files can be specified in a single destination definition, as needed: ■ To add an additional input database file to the definition, click **New**. A blank row in the table of input database files appears. Fill in the blank row. ■ To delete an input database ID from the definition, click on the check box in the corresponding **All** column and then click **Delete**. To delete all database files from the list, click the **All** check box in the column heading and then click **Delete**. The input database file is removed from the list of input database files. Only unique combinations of Input DBID and Input File parameters can be specified in a single Adabas destination definition. This implies that the input from a database and file combination can only have a single Adabas target within a specific Adabas destination definition. If you want the input from a database and file combination to go to more than one Adabas target, define multiple Adabas destination definitions.	---
Input File (DAIFILE)	The number of the input file for this Adabas destination. The input database ID and file specified must also be included in an SFILE definition in the subscription. At least one file must be listed for an Adabas destination definition. Multiple input database files can be specified in a single destination definition, as needed: ■ To add an additional input database file to the definition, click **New**. A blank row in the table of input database files appears. Fill in the blank row. ■ To delete an input database ID from the definition, click on the check box in the corresponding **All** column and then click **Delete**. To delete all database files from the list, click the **All** check box in the column heading and then click **Delete**. The input database file is removed from the list of input database files. Only unique combinations of Input DBID and Input File parameters can be specified in a single Adabas destination definition. This implies that the input from a database and file combination can only have a single Adabas target within a specific Adabas destination definition. If you want the input from a database and file combination to go to more than one Adabas target, define multiple Adabas destination definitions.	---
Target DBID (DATDBID)	The database ID associated with the target file for the replicated data. The database ID is numeric and can range from one to 65535. Multiple target database IDs and files can be specified in a single destination definition, as needed.	---

Parameter Name	Specify	Default
	■ To add an additional target database file to the definition, click **New**. A blank row in the table of target database files appears. Fill in the blank row. ■ To delete a target database ID from the definition, click on the check box in the corresponding **All** column and then click **Delete**. To delete all database files from the list, click the **All** check box in the column heading and then click **Delete**. The target database file is removed from the list of target database files.	
Target File (DATFILE)	The number of the target (output) file for the replicated data associated with the input file in this destination definition. There is no default. Multiple target database IDs and files can be specified in a single destination definition, as needed. ■ To add an additional target database file to the definition, click **New**. A blank row in the table of target database files appears. Fill in the blank row. ■ To delete a target database ID from the definition, click on the check box in the corresponding **All** column and then click **Delete**. To delete all database files from the list, click the **All** check box in the column heading and then click **Delete**. The target database file is removed from the list of target database files.	---
⊘	This button allows you to supply additional parameter specifications for each input and target DBID/file pair listed in this Adabas destination definition. The description of these additional parameters is provided in the **later steps** of this procedure.	

2 If you want to add additional input-target destination file pairings, click the **New** button to add new lines to the list of input-target file pairings. Fill in the fields for each line as described in the previous step.

3 If you want to delete an input-target file pairing from the list, click in the check box in the **All** column corresponding to the line for the pairing you want to delete and click the **Delete** button.

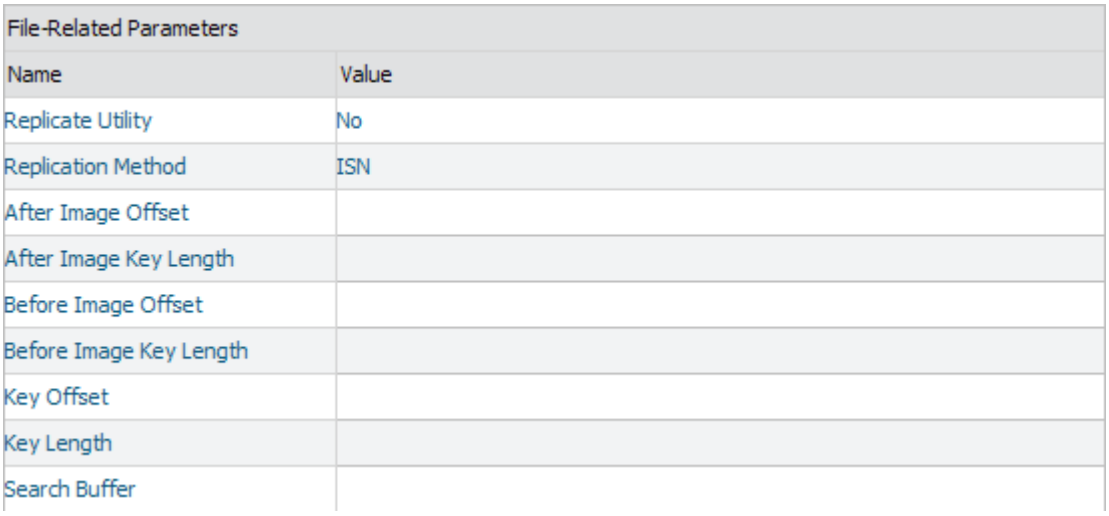
The file pairing is deleted from the list.

Step 4. (Optional) Specify File-Related Parameters for the Adabas Destination

To use Event Replicator Administration to specify file-related parameters for the Adabas destination, complete the following steps:

1 On the **Adabas Destination** panel, click on the ⊘ button to the right of the input-target file pairing for which you want to specify file-related parameter settings.

A new panel appears allowing you to specify additional parameters for the input/target DBID/file pair. The fields for file-related parameter settings appears at the bottom of this panel:



2 Fill in the fields in **Value** columns for the file-related parameters, as described in the following table:

Parameter Name	Description	Default
Replicate Utility (DAREPLICATEUTI)	This parameter can be specified regardless of the Replication Method selected. Specify whether Adabas utility change replication should be activated for a specific target file at Event Replicator Server startup. Valid values are "Yes" and "No". If "Yes" is specified, utility replication is activated for the target file at Event Replicator Server startup; if "No" is specified, utility replication is not activated for the target file. For more information about replicating utility functions, read *Replicating Utility Functions*, in *Event Replicator for Adabas Concepts*.	No
Replication Method (DATMETHOD)	Specify the method to be used when searching for a record on the target database. Valid values are "ISN" and "KEY". When this parameter is set to "KEY", the parameters Search Buffer, After Image Offset, After Image Key Length, Before Image Offset, Before Image Key Length, Key Offset, and Key Length may also be supplied to indicate where to find the key in the replicated data. All of these parameters are optional except Search Buffer and After Image Length, which are required.	ISN

Parameter Name	Description	Default
After Image Offset (DATKEYAIO)	This optional parameter can only be set if the Replication Method parameter is set to "KEY"; if the Replication Method parameter is set to ISN, this parameter may not be specified. Specify the offset of the key to be used in the after image of the data buffer. Valid values range from 0 through 2,147,483,646 bytes. The sum of this parameter value and the length of the key to be used in the after image (After Image Key Length parameter), must be less than or equal to 2,147,483,647 bytes.	0
After Image Key Length (DATKEYAIL)	When the Replication Method parameter is set to "KEY", this parameter is required. If the Replication Method parameter is set to ISN, this parameter may not be specified. Specify the length (in bytes) of the key to be used in the after image of the data buffer. Valid values range from 1 through 32,767 bytes.	none
Before Image Offset (DATKEYBIO)	This optional parameter can only be set if the Replication Method parameter is set to "KEY"; if the Replication Method parameter is set to ISN, this parameter may not be specified. Specify the offset of the key to be used in the before image of the data buffer. Valid values range from 0 through 2,147,483,646 bytes. The sum of this parameter value and the length of the key to be used in the before image (Before Image Key Length parameter), must be less than or equal to 2,147,483,647 bytes. If this parameter is specified, the Before Image Key Length parameter must also be specified with a value greater than zero.	0
Before Image Key Length (DATKEYBIL)	This optional parameter can only be set if the Replication Method parameter is set to "KEY"; if the Replication Method parameter is set to ISN, this parameter may not be specified. If the Before Image Offset parameter is specified, this parameter must also be specified with a value greater than zero. Specify the length (in bytes) of the key to be used in the before image of the data buffer. Valid values range from 1 through 32767 bytes.	none
Key Offset (DATKEYKYO)	This optional parameter can only be set if the Replication Method parameter is set to "KEY"; if the Replication Method parameter is set to ISN, this parameter may not be specified. Specify the offset of the key to be used in the before image of the primary key. Valid values range from 0 through 32767 bytes. The sum of this parameter value and the length of the key to be used in the before image of the primary key (Key Length parameter), must be less than or equal to 32767 bytes. If this parameter is specified, the Key Length parameter must also be specified with a value greater than zero.	0

Parameter Name	Description	Default
Key Length (DATKEYKYL)	This optional parameter can only be set if the Replication Method parameter is set to "KEY"; if the Replication Method parameter is set to ISN, this parameter may not be specified. If the Key Offset parameter is specified, this parameter must also be specified with a value greater than zero. Specify the length (in bytes) of the key to be used in the before image of the primary key. Valid values range from 1 through 32767 bytes.	none
Search Buffer (DATKEYSB)	When the Replication Method parameter is set to "KEY", this parameter is required. If the Replication Method parameter is set to ISN, this parameter may not be specified. Specify a search buffer to be used for keyed replication. Up to 60 alphanumeric characters can be specified.	none

3 When you are satisfied with the file-related parameters pairings listed on the panel, click the **OK** button to return to the first panel of Adabas destination definitions.

Step 5. Save the Adabas Destination Definition

To use Event Replicator Administration to save an Adabas destination definition:

- When all specifications have been made to your satisfaction, click **OK** on the Adabas destination panel.

  The Adabas destination definition is saved in the Replicator system file.

Creating a webMethods EntireX Destination Definition

Using a webMethods EntireX destination definition, replicated data is written to an output queue via webMethods EntireX. Prior to using webMethods EntireX as the messaging subsystem, be sure to read *Using webMethods EntireX as the Messaging System*, in *Event Replicator for Adabas Administration and Operations Guide* provided with your Event Replicator Administration documentation.

To create a webMethods EntireX destination definition in Event Replicator Administration, complete the following steps:

Step 1. Access the webMethods EntireX Destination Definition Creation Area

To access the webMethods EntireX destination definition creation area of Event Replicator Administration:

1 List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*, elsewhere in this guide.

The destination definitions are listed in detail-view.

2 Right-click on **Destinations** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

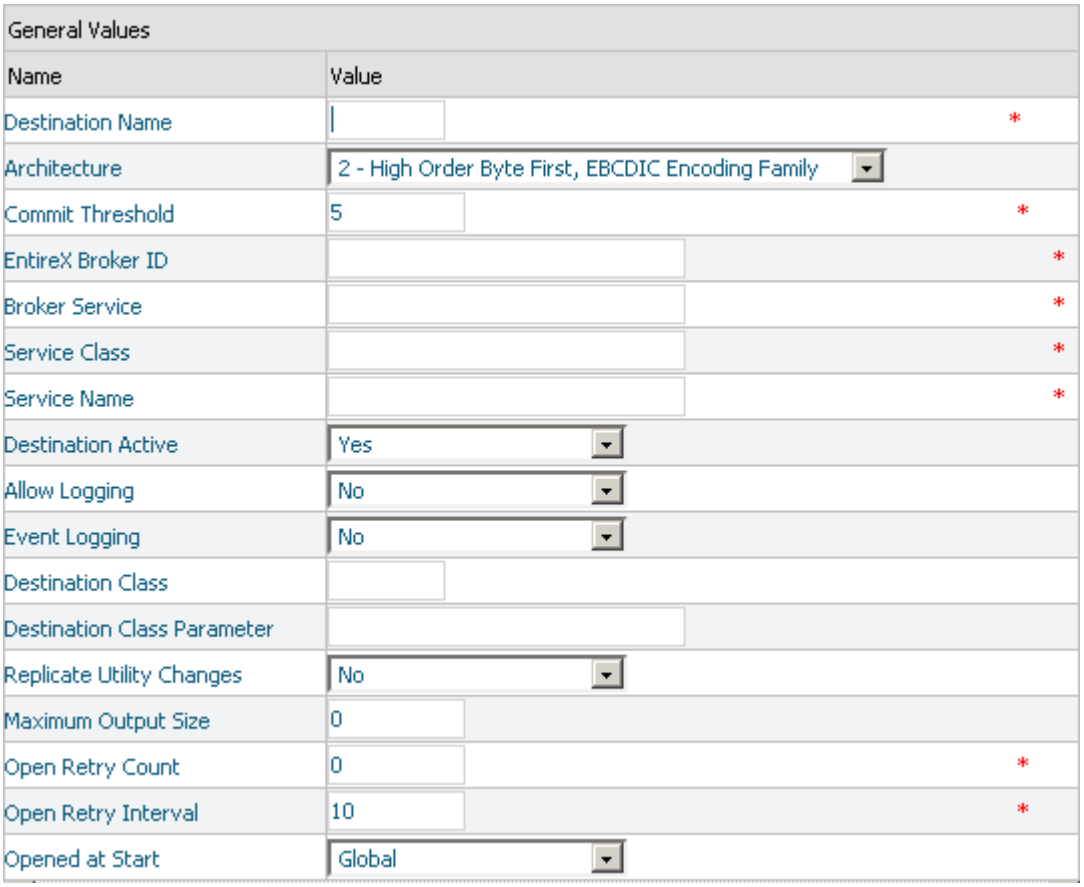
3 Click on **Create New EntireX Communicator Destination** in the drop-down menu.

A blank **EntireX Communicator Destination** panel appears in detail-view.

Step 2. Specify General and TLOG webMethods EntireX Destination Parameters

To use Event Replicator Administration to supply general and TLOG specifications for a webMethods EntireX destination definition, complete the following steps:

1 Supply general parameter specifications for the webMethods EntireX destination. The fields for these general parameters can be found at the top of the **EntireX Communicator Destination** panel:



Supply values for the these fields, as described in the following table:

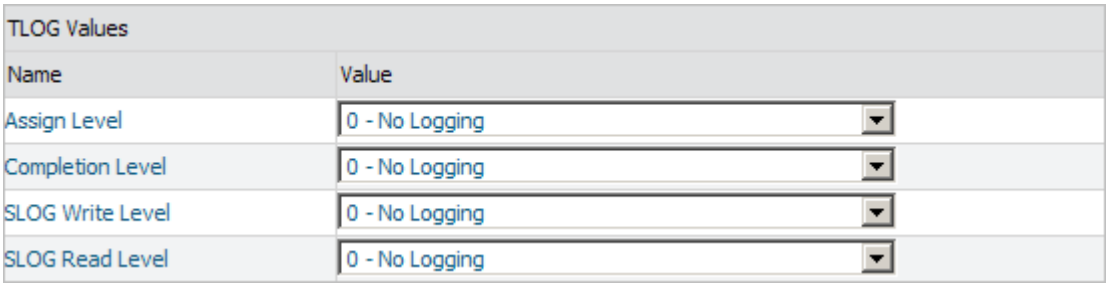
Parameter Name	Specify	Default
Destination Name (`DESTINATION NAME`)	The unique name for the webMethods EntireX destination definition. The specified name must be alphanumeric and be between one and eight characters long.	---
Architecture (`DARC`)	The data architecture for fields in the URB* control structures sent to the webMethods EntireX destination. For complete information on calculating a value for this parameter, read about the DARC parameter in *Event Replicator for Adabas Reference Guide* provided with your Event Replicator Administration documentation.	2 (High-order byte first, EBCDIC encoding)
Commit Threshold (`DCOMMITTHRESHOLD`)	The number of messages that will be sent to the webMethods EntireX destination before a commit is performed for those messages.	5

Parameter Name	Specify	Default
EntireX Broker ID (DETBBROKERID)	The webMethods EntireX Broker ID for which this destination definition applies. The name can be up to 32 characters long. Broker IDs come in two formats: one for TCP/IP communications and one for Adabas SVC communications. For TCP/IP communications, the format is: *addr*:*port-number*:TCP In this case, the *addr* setting is either the TCP/IP address or the host name. The *port-number* setting should match the webMethods EntireX PORT parameter. For Adabas SVC communications, the format is: '*broker-id*:SVC*nnn*:NET' In this case, the *broker-id* setting should match the webMethods EntireX BROKER-ID parameter in the Broker ETBFILE DD. The *nnn* setting should match either the webMethods EntireX ADASVC or ADA5SVC parameters in the Broker PARMS DD statement. If no name is specified, the default webMethods EntireX Broker ID specified by the ETBBROKERID parameter is used.	---
Broker Service (DETBSERVICE)	The webMethods EntireX Broker service for which this destination definition applies. The service identification can be up to 32 characters long.	---
Service Class (DETBSERVICECLASS)	The webMethods EntireX Broker service class name for which this destination definition applies. The name can be up to 32 characters long.	---
Service Name (DETBSERVICENAME)	The webMethods EntireX service name for which the webMethods EntireX destination definition applies. The name can be up to 32 characters long.	---
Destination Active (DACTIVE)	Whether or not this destination definition should be activated for use once it is loaded by the Event Replicator Server. Valid values are "Yes" (load and activate the definition) or "No" (load, but do not activate the definition).	Yes
Allow Logging (DLOG)	Whether or not subscription logging should be activated for this destination definition. Valid values are "Yes" (activate subscription logging) or "No" (do not activate subscription logging).	No
Event Logging (*DEVENTLOG*)	Whether or not events should be logged by the Event Replicator Server and sent to this destination. This is the equivalent of specifying the *DEVENTLOG* directly in the Event Replicator Server startup job. Valid values are "Yes" or "No". When this optional parameter is set to "Yes", Event Replicator Server events are logged to the destination. When this parameter is set to "No" (the default), they are not.	No

Parameter Name	Specify	Default
	Event Replicator Server events are logged in URBS elements. In releases prior to 3.2.1, these URBS elements were sent to destinations related to the event itself. Starting with release 3.2.1, the URBS elements are also sent to any other destinations you have defined "Event Logging =Yes". If a related destination also is defined with "Event Logging =Yes", it will only receive one instance of the URBS element. To access this log of Event Replicator Server events in the destination queue, you must supply your own application that reads the event URBS elements in the destination queue. If such an application does not exist, the logged events simply sit in the queue.	
Destination Class (DCLASS)	For the moment, leave this field blank. It is described later in this section.	---
Destination Class Parameter (DCLASSPARM)	For the moment, leave this field blank. It is described later in this section.	---
Replicate Utility Changes (DREPLICATEUTI)	Whether Adabas utility change replication should be activated for a destination at Event Replicator Server startup. Valid values are "Yes" and "No". If "Yes" is specified, utility replication is activated for the destination at Event Replicator Server startup; if "No" is specified, utility replication is not activated for the destination. For more information about replicating utility functions, read *Replicating Utility Functions*, in *Event Replicator for Adabas Concepts*.	No
Maximum Output Size (DMAXOUTPUTSIZE parameter)	The maximum output size (in bytes) for the destination. This is the equivalent of specify the DMAXOUTPUTSIZE parameter directly in the Event Replicator Server startup job. Valid values are 0 or any integer ranging from 4096 through 2,147,483,647. You can specify the value for this parameter in a purely numeric form or use K at the end of the number to specify kilobytes. For example, DMAXOUTPUTSIZE=4K is the same as DMAXOUTPUTSIZE=4096. The value for this parameter will be used if it is less than or equal to the maximum output size for the Event Replicator Server (specified using the MAXOUTPUTSIZE global parameter) and less than or equal to the maximum output allowed for the messaging system queue being defined. If this value is larger than the MAXOUTPUTSIZE specification or the maximum output size allowed by the messaging system, the smaller value will be used. A value of 0 indicates that no specific limit is set for this destination. Instead, the smaller of the MAXOUTPUTSIZE	0

Parameter Name	Specify	Default
	specification or the maximum output size allowed by the messaging system will be used.	
Open Retry Count (DRETRYCOUNT parameter)	The number of times that an attempt to open the destination will be retried at the interval specified by the **Retry Interval parameter**. This is the equivalent of specifying the DRETRYCOUNT parameter directly in the Event Replicator Server startup job. Valid values range from 0 through 2,147,483,647 or the literal "GLOBAL". If the value "GLOBAL" is specified for this parameter, the specification for the **Retry Count global variable** will be used. Any retry attempts will occur at the interval specified by the **Retry Interval** parameter. A value of zero indicates that no retry attempt to open this destination should occur.	The value of the **Retry Count global variable**
Open Retry Interval (DRETRYINTERVAL parameter)	The default number of seconds between retry attempts to open the destination. This is the equivalent of specifying the DRETRYINTERVAL parameter directly in the Event Replicator Server startup job. Valid values are 0, 5 through 2,147,483,647, or the literal "GLOBAL". If the value "GLOBAL" is specified for this parameter, the specification for the **Retry Interval global variable** will be used. A value of zero indicates that no retry attempt to open this destination should occur. Except for a specification of zero, the minimum value that can be specified for this parameter is 5 seconds.	The value of the **Retry Interval global variable**.
Opened at Start (DOPEN parameter)	Whether or not the destination should be opened at Event Replicator Server startup. Valid values are "Yes", "No", or "Global", with "Global" as the default. When this parameter is set to "Yes", the destination is opened at Event Replicator Server startup. When this parameter is set to "No", the destination is *not* opened at Event Replicator Server startup. When this parameter is set to "Global", the decision to open the destination at Event Replicator Server startup depends on the setting of the **Global OPEN Value** (GOPEN) global parameter. If GOPEN=YES, the destination is opened at Event Replicator Server startup; if GOPEN=NO, it is not opened. This is the equivalent of specifying the DOPEN parameter in the Event Replicator Server startup job.	Global

2 Supply TLOG parameter specifications for the webMethods EntireX destination. The fields for these TLOG parameters can be found at the bottom of the **EntireX Communicator Destination** panel:



Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Assign Level (`DTLASSIGN`)	The level of transaction logging that should occur when a transaction is assigned to a destination for output processing. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Completion Level (`DTLCOMP`)	The level of transaction logging that should occur when a transaction has been successfully output to the messaging system. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Write Level (`DTLSLOGWRITE`)	The level of transaction logging that should occur when a transaction has been successfully written to the SLOG file. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Read Level (`DTLSLOGREAD`)	The level of transaction logging that should occur when a transaction has been successfully read from the SLOG and is about to be queued for output to the destination. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging

Step 3. (Optional) Specify Destination Class Information, If Applicable

The Destination Class and Destination Class Parameter fields can be used to invoke and pass parameters to the Event Replicator Target Adapter for this destination. The fields are located in the middle of the general fields for the webMethods EntireX destination. Do not use these fields unless you want to invoke and pass parameters to the Event Replicator Target Adapter for the destination or unless otherwise requested by a Software AG support representative.

If applicable, use the Destination Class (`DCLASS`) field to specify the destination class for this destination definition. Valid values are "NONE" (interpreted as a blank) or "SAGTARG" (if Event Replicator Target Adapter processing should be invoked). There is no default.

If you specify a value for the Destination Class field, you can optionally use the Destination Class Parameter (`DCLASSPARM`) field to specify up to 120 bytes of character data to be passed to the optional destination output user exit.

If DCLASS=SAGTARG is specified (if the Destination Class field is set to "SAGTARG") to invoke the Event Replicator Target Adapter for a destination, you may want to specify one or more of the following keyword parameters:

Note: These parameter keywords must be specified in uppercase.

NOSPRE

Specify the "NOSPRE" keyword in the DCLASSPARM parameter if you do not want the subscription name to prefix the names of the tables produced by the Event Replicator Target Adapter. When "NOSPRE" is specified, the schema file name (Predict view name) alone is used for the table names; when "NOSPRE" is *not* specified, the subscription name prefixes the schema file name in the table names.

Note: Oracle identifiers are limited to 30 characters. If NOSPRE is *not* specified and an Oracle RDBMS is used by the Event Replicator Target Adapter, the identifier names may exceed 30 characters and errors may occur. We recommend using NOSPRE if an Oracle RDBMS is also used.

OPTIONS

The OPTIONS keyword parameter can be used to specify options for the destination. Specify one or more of the options described in the following table, using the syntax `OPTIONS=val1, val2,...`

Option Value	Description
1	This option is no longer supported. If specified, it will be ignored.
2	Specify OPTIONS=2 to indicate that long names should be used. This option will cause long names to be sent in place of the default short names used for various elements and attributes. Short names are the default and save on the amount of data being transferred. Long names make for better readability. For example, the short name <F> would appear as <Field> using long names.
4	Specify OPTIONS=4 to ensure that invalid XML characters found in alphanumeric fields are not translated to spaces.
8	Specify OPTIONS=8 to ensure that trailing blanks in alphanumeric fields are not removed.
16	Specify OPTIONS=16 to ensure that characters used by XML are not replaced automatically with predefined entity references. For example, if OPTIONS=16 is set, the ampersand (&) character would not be replaced with the literal "&".
32	Specify OPTIONS=32 to send the full image on an update. The full before image (if available) and after image of all fields are sent for an update, even if the field values were not changed or are null.

TRACE

The TRACE keyword parameter can be used to specify the contents of the trace. Specify the TRACE keyword parameter using the syntax `TRACE=nnnn`. Possible values of the TRACE (*nnnn*) are listed in the following table. However, if you want to trace multiple control blocks, add their trace values together and enter the total value. For example, to trace the before and after images of the URBD control blocks, you would specify `TRACE=24` because the sum of 8 (URBD control block before image) and 16 (URBD control block after image) is 24.

Note: If tracing is enabled using this keyword parameter, be sure to include the following JCL statement in the startup JCL of the Event Replicator Server:

```
//DDTRACE1 DD SYSOUT=X
```

Trace Value	Description
1	Trace the URBS control block.
2	Trace the URBT control block.
4	Trace the URBR control block.
8	Trace the URBD control block before image.
16	Trace the URBD control block after image.
32	Trace the URBF/URBG control block before image.
64	Trace the URBF/URBG control block after image.
128	Trace the send buffer.
256	Trace the URBY control block.
512	Trace the URBO control block.
1024	Trace the output parameters.

Trace Value	Description
2048	Trace the subscription table.

Step 4. Save the webMethods EntireX Destination Definition

To use Event Replicator Administration to save a webMethods EntireX destination definition:

- When all specifications have been made to your satisfaction, click **OK** on the webMethods EntireX destination panel.

  The webMethods EntireX destination definition is saved in the Replicator system file.

Creating a WebSphere MQ Destination Definition

Using a WebSphere MQ destination definition, replicated data is written to an output queue via IBM WebSphere MQ. Prior to using WebSphere MQ as the messaging subsystem, be sure to read *Using WebSphere MQ as the Messaging System*, in *Event Replicator for Adabas Administration and Operations Guide* provided with your Event Replicator Administration documentation.

To create a WebSphere MQ destination definition in Event Replicator Administration, complete the following steps:

Step 1. Access the WebSphere MQ Destination Definition Creation Area

To access the WebSphere MQ destination definition creation area of Event Replicator Administration:

1 List the destination definitions in Event Replicator Administration, as described in ***Listing Destination Definitions***, elsewhere in this guide.

  The destination definitions are listed in detail-view.

2 Right-click on **Destinations** in the tree-view under **Replication Definitions**.

  A drop-down menu appears.

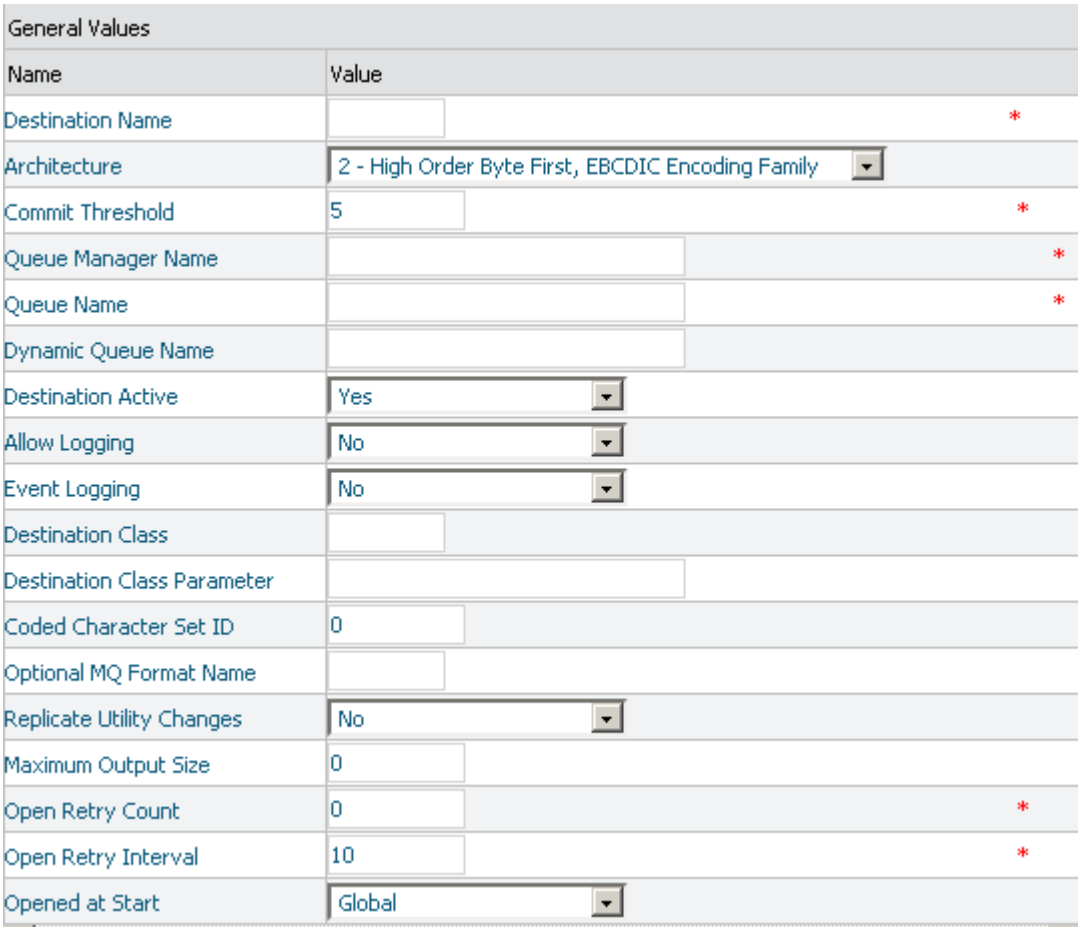
3 Click on **Create New MQSeries Destination** in the drop-down menu.

  A blank **MQSeries Destination** panel appears in detail-view.

Step 2. Specify General and TLOG WebSphere MQ Destination Parameters

To use Event Replicator Administration to supply general and TLOG specifications for a WebSphere MQ destination definition, complete the following steps:

1 Supply general parameter specifications for the WebSphere MQ destination. The fields for these general parameters can be found at the top of the **MQSeries Destination** panel:



Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Destination Name (NAME)	The unique name for the WebSphere MQ destination definition. The specified name must be alphanumeric and be between one and eight characters long.	---

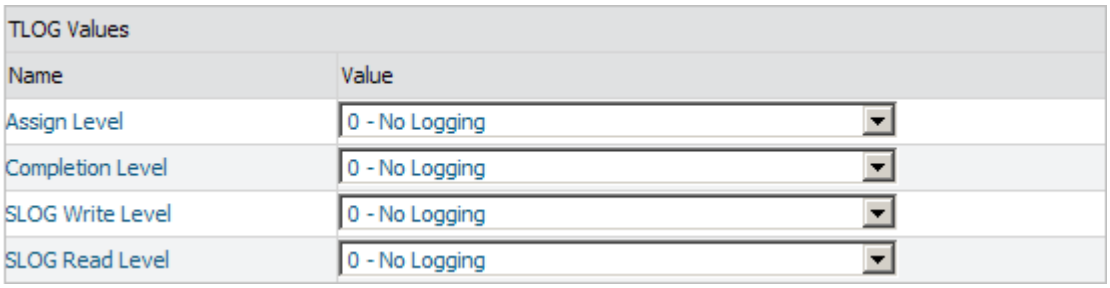
Parameter Name	Specify	Default
Architecture (DARC)	The data architecture for fields in the URB* control structures sent to the WebSphere MQ destination. For complete information on calculating a value for this parameter, read *DARC*, in *Event Replicator for Adabas Reference Guide* provided with your Event Replicator Administration documentation.	2 (High-order byte first, EBCDIC encoding)
Commit Threshold (DCOMMITTHRESHOLD)	The number of messages that will be sent to the WebSphere MQ destination before a commit is performed for those messages. The term "commit" in this context means that the Event Replicator Server informs the messaging system that all messages sent (since the last commit) should be made permanent. In the case of WebSphere MQ, commit means that the Event Replicator Server will issue an MQCMIT call for the queue.	5
Queue Manager Name (DMQQMGRNAME)	The WebSphere MQ queue manager name. The name can be up to 48 characters long.	---
Queue Name (DMQQNAME)	The WebSphere MQ queue name. The name can be up to 48 characters long. There is no default.	---
Dynamic Queue Name (DMQDYNQNAME)	The WebSphere MQ dynamic queue name. The name can be up to 48 characters long.	blanks
Destination Active (DACTIVE)	Whether or not this destination definition should be activated for use once it is loaded by the Event Replicator Server. Valid values are "Yes" (load and activate the definition) or "No" (load, but do not activate the definition).	Yes
Allow Logging (DLOG)	Whether or not subscription logging should be activated for this destination definition. Valid values are "Yes" (activate subscription logging) or "No" (do not activate subscription logging).	No
Event Logging (*DEVENTLOG*)	Whether or not events should be logged by the Event Replicator Server and sent to this destination. This is the equivalent of specifying the *DEVENTLOG* directly in the Event Replicator Server startup job. Valid values are "Yes" or "No". When this optional parameter is set to "Yes", Event Replicator Server events are logged to the destination. When this parameter is set to "No" (the default), they are not. Event Replicator Server events are logged in URBS elements. In releases prior to 3.2.1, these URBS elements were sent to destinations related to the event itself. Starting with release 3.2.1, the URBS elements are also sent to any other destinations you have defined "Event Logging =Yes". If a related destination also is defined with "Event Logging =Yes", it will only receive one instance of the URBS element. To access this log of Event Replicator Server events in the destination queue, you must supply your own application that	No

Parameter Name	Specify	Default
	reads the event URBS elements in the destination queue. If such an application does not exist, the logged events simply sit in the queue.	
Destination Class (DCLASS)	For the moment, leave this field blank. It is described later in this section.	---
Destination Class Parameter (DCLASSPARM)	For the moment, leave this field blank. It is described later in this section.	---
Coded Character Set ID (DMQCCSID)	The destination-specific coded character set ID (CCSID) for the WebSphere MQ destination. This is the equivalent of specifying the DMQCCSID parameter in the Event Replicator Server startup job. Valid values range from 0 through 2,147,483,647. This optional parameter can only be specified when the DCLASS or DEXIT parameters are specified. The Event Replicator Server does not attempt to verify the value of this parameter as the character codes may be changed or added to as time goes on. The value for this parameter is simply passed in the appropriate WebSphere MQ request as the CCSID.	0
Optional MQ Format Name (DMQFORMAT)	The optional MQ format name. The format name can be up to eight characters long. **Note:** You cannot specify a value for this parameter if a value has not also been specified for the Destination Class parameter.	---
Replicate Utility Changes (DREPLICATEUTI)	Whether Adabas utility change replication should be activated for a destination at Event Replicator Server startup. Valid values are "Yes" and "No". If "Yes" is specified, utility replication is activated for the destination at Event Replicator Server startup; if "No" is specified, utility replication is not activated for the destination. For more information about replicating utility functions, read *Replicating Utility Functions*, in *Event Replicator for Adabas Concepts*.	No
Maximum Output Size (DMAXOUTPUTSIZE)	The maximum output size (in bytes) for the destination. This is the equivalent of specify the DMAXOUTPUTSIZE parameter directly in the Event Replicator Server startup job. Valid values are 0 or any integer ranging from 4096 through 2,147,483,647. You can specify the value for this parameter in a purely numeric form or use K at the end of the number to specify kilobytes. For example, DMAXOUTPUTSIZE=4K is the same as DMAXOUTPUTSIZE=4096. The value for this parameter will be used if it is less than or equal to the maximum output size for the Event Replicator Server (specified using the MAXOUTPUTSIZE global parameter) and less	0

Parameter Name	Specify	Default
	than or equal to the maximum output allowed for the messaging system queue being defined. If this value is larger than the MAXOUTPUTSIZE specification or the maximum output size allowed by the messaging system, the smaller value will be used. A value of 0 indicates that no specific limit is set for this destination. Instead, the smaller of the MAXOUTPUTSIZE specification or the maximum output size allowed by the messaging system will be used.	
Open Retry Count (DRETRYCOUNT)	The number of times that an attempt to open the destination will be retried at the interval specified by the **Retry Interval parameter**. This is the equivalent of specifying the DRETRYCOUNT parameter directly in the Event Replicator Server startup job. Valid values range from 0 through 2,147,483,647 or the literal "GLOBAL". If the value "GLOBAL" is specified for this parameter, the specification for the **Retry Count global variable** will be used. Any retry attempts will occur at the interval specified by the **Retry Interval** parameter. A value of zero indicates that no retry attempt to open this destination should occur.	The value of the **Retry Count global variable**
Open Retry Interval (DRETRYINTERVAL)	The default number of seconds between retry attempts to open the destination. This is the equivalent of specifying the DRETRYINTERVAL parameter directly in the Event Replicator Server startup job. Valid values are 0, 5 through 2,147,483,647, or the literal "GLOBAL". If the value "GLOBAL" is specified for this parameter, the specification for the **Retry Interval global variable** will be used. A value of zero indicates that no retry attempt to open this destination should occur. Except for a specification of zero, the minimum value that can be specified for this parameter is 5 seconds.	The value of the **Retry Interval global variable**
Opened at Start (DOPEN parameter)	Whether or not the destination should be opened at Event Replicator Server startup. Valid values are "Yes", "No", or "Global", with "Global" as the default. When this parameter is set to "Yes", the destination is opened at Event Replicator Server startup. When this parameter is set to "No", the destination is *not* opened at Event Replicator Server startup. When this parameter is set to "Global", the decision to open the destination at Event Replicator Server startup depends on the setting of the **Global OPEN Value** (GOPEN) global parameter.	Global

Parameter Name	Specify	Default
	If GOPEN=YES, the destination is opened at Event Replicator Server startup; if GOPEN=NO, it is not opened. This is the equivalent of specifying the DOPEN parameter in the Event Replicator Server startup job.	

2 Supply TLOG parameter specifications for the WebSphere MQ destination. The fields for these TLOG parameters can be found at the bottom of the **MQSeries Destination** panel:



Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Assign Level (DTLASSIGN)	The level of transaction logging that should occur when a transaction is assigned to a destination for output processing. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Completion Level (DTLCOMP)	The level of transaction logging that should occur when a transaction has been successfully output to the messaging system. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Write Level (DTLSLOGWRITE parameter)	The level of transaction logging that should occur when a transaction has been successfully written to the SLOG file. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Read Level (DTLSLOGREAD parameter)	The level of transaction logging that should occur when a transaction has been successfully read from the SLOG and is about to be queued for output to the destination. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging

Step 3. (Optional) Specify Destination Class Information, If Applicable

The Destination Class and Destination Class Parameter fields can be used to invoke and pass parameters to the Event Replicator Target Adapter for this destination. The fields are located in the middle of the general fields for the WebSphere MQ destination. Do not use these fields unless you want to invoke and pass parameters to the Event Replicator Target Adapter for the destination or unless otherwise requested by a Software AG support representative.

If applicable, use the Destination Class (`DCLASS`) field to specify the destination class for this destination definition. Valid values are "NONE" (interpreted as a blank) or "SAGTARG" (if Event Replicator Target Adapter processing should be invoked). There is no default.

If you specify a value for the Destination Class field, you can optionally use the Destination Class Parameter (`DCLASSPARM`) field to specify up to 120 bytes of character data to be passed to the optional destination output user exit.

If DCLASS=SAGTARG is specified (if the Destination Class field is set to "SAGTARG") to invoke the Event Replicator Target Adapter for a destination, you may want to specify one or more of the following keyword parameters:

Note: These parameter keywords must be specified in uppercase.

NOSPRE

Specify the "NOSPRE" keyword in the DCLASSPARM parameter if you do not want the subscription name to prefix the names of the tables produced by the Event Replicator Target Adapter. When "NOSPRE" is specified, the schema file name (Predict view name) alone is used for the table names; when "NOSPRE" is *not* specified, the subscription name prefixes the schema file name in the table names.

Note: Oracle identifiers are limited to 30 characters. If NOSPRE is *not* specified and an Oracle RDBMS is used by the Event Replicator Target Adapter, the identifier names may exceed 30 characters and errors may occur. We recommend using NOSPRE if an Oracle RDBMS is also used.

OPTIONS

The OPTIONS keyword parameter can be used to specify options for the destination. Specify one or more of the options described in the following table, using the syntax `OPTIONS=`*`val1`*`,`*`val2`*`,...`

Option Value	Description
1	This option is no longer supported. If specified, it will be ignored.
2	Specify OPTIONS=2 to indicate that long names should be used. This option will cause long names to be sent in place of the default short names used for various elements and attributes. Short names are the default and save on the amount of data being transferred. Long names make for better readability. For example, the short name <F> would appear as <Field> using long names.
4	Specify OPTIONS=4 to ensure that invalid XML characters found in alphanumeric fields are not translated to spaces.
8	Specify OPTIONS=8 to ensure that trailing blanks in alphanumeric fields are not removed.
16	Specify OPTIONS=16 to ensure that characters used by XML are not replaced automatically with predefined entity references. For example, if OPTIONS=16 is set, the ampersand (&) character would not be replaced with the literal "&".
32	Specify OPTIONS=32 to send the full image on an update. The full before image (if available) and after image of all fields are sent for an update, even if the field values were not changed or are null.

TRACE

The TRACE keyword parameter can be used to specify the contents of the trace. Specify the TRACE keyword parameter using the syntax `TRACE=nnnn`. Possible values of the TRACE (*nnnn*) are listed in the following table. However, if you want to trace multiple control blocks, add their trace values together and enter the total value. For example, to trace the before and after images of the URBD control blocks, you would specify `TRACE=24` because the sum of 8 (URBD control block before image) and 16 (URBD control block after image) is 24.

Note: If tracing is enabled using this keyword parameter, be sure to include the following JCL statement in the startup JCL of the Event Replicator Server:

```
//DDTRACE1 DD SYSOUT=X
```

Trace Value	Description
1	Trace the URBS control block.
2	Trace the URBT control block.
4	Trace the URBR control block.
8	Trace the URBD control block before image.
16	Trace the URBD control block after image.
32	Trace the URBF/URBG control block before image.
64	Trace the URBF/URBG control block after image.
128	Trace the send buffer.
256	Trace the URBY control block.
512	Trace the URBO control block.
1024	Trace the output parameters.

Trace Value	Description
2048	Trace the subscription table.

Step 4. Save the WebSphere MQ Destination Definition

To use Event Replicator Administration to save a WebSphere MQ destination definition:

- When all specifications have been made to your satisfaction, click **OK** on the WebSphere MQ destination panel.

  The WebSphere MQ destination definition is saved in the Replicator system file.

Creating a File Destination Definition

Using a File destination definition, replicated data is written to the CLOG, using TLOG URBLTDOD records. You can use these records in the CLOG file to create a sequential output file of the replicated data. For more information, read *Creating a Sequential Output File* in *Event Replicator for Adabas Administration and Operations Guide* provided with your Event Replicator Administration documentation.

Caution: Be sure that the CLOG is defined in the Event Replicator Server startup JCL (via one or more DDCLOGR*n* DD statements) if you will be using a File destination definition during Event Replicator for Adabas processing. If you do not, a warning message will be issued and the File destination will be set to "Unavailable". For more information about the CLOG, read your Adabas documentation.

To create a File destination definition in Event Replicator Administration, complete the following steps:

Step 1. Access the File Destination Definition Creation Area

To access the File destination definition creation area of Event Replicator Administration:

1 List the destination definitions in Event Replicator Administration, as described in ***Listing Destination Definitions***, elsewhere in this guide.

  The destination definitions are listed in detail-view.

2 Right-click on **Destinations** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

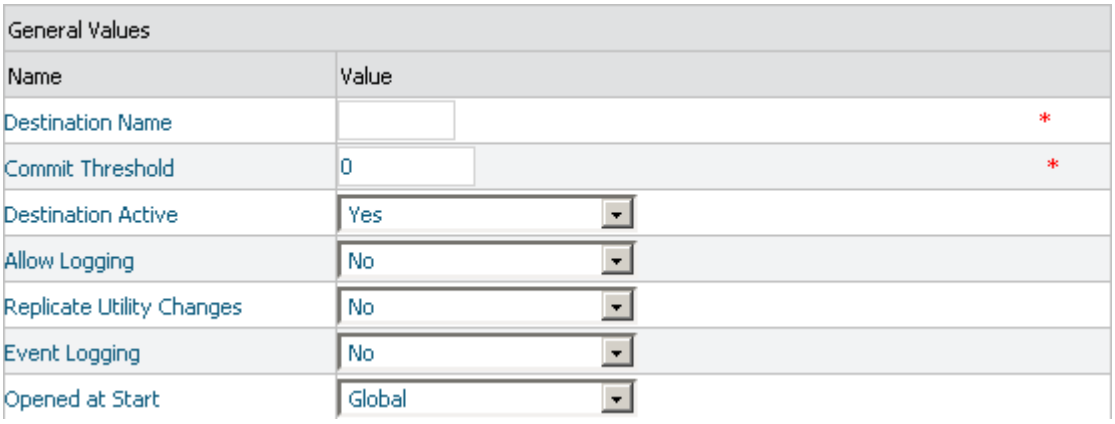
3 Click on **Create New File Destination** in the drop-down menu.

A blank **File Destination** panel appears in detail-view.

Step 2. Specify General and TLOG File Destination Parameters

To use Event Replicator Administration to supply general and TLOG specifications for a File destination definition, complete the following steps:

1 Supply general parameter specifications for the File destination. The fields for these general parameters can be found at the top of the **File Destination** panel:



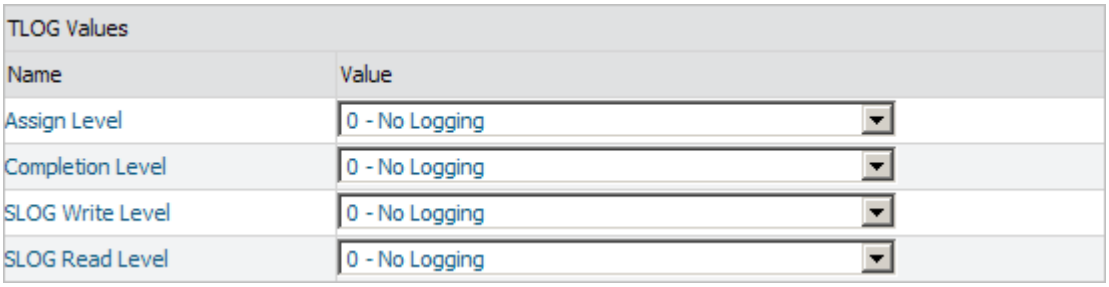
Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Destination Name (NAME)	The unique name for the File destination definition. The specified name must be alphanumeric and be between one and eight characters long.	---
Commit Threshold (DCOMMITTHRESHOLD)	The number of URBLTDOD TLOG record bytes that will be written to the CLOG sequential file before the buffers are flushed. The term "commit" in this context means that the Event Replicator Server informs the messaging system that all messages sent (since the last commit) should be made permanent. In the case of WebSphere MQ, commit means that the Event Replicator Server will issue an MQCMIT call for the queue.	5
Destination Active (DACTIVE)	Whether or not this destination definition should be activated for use once it is loaded by the Event Replicator Server. Valid values are "Yes" (load and activate the definition) or "No" (load, but do not activate the definition).	Yes

Parameter Name	Specify	Default
Allow Logging (DLOG)	Whether or not subscription logging should be activated for this destination definition. Valid values are "Yes" (activate subscription logging) or "No" (do not activate subscription logging).	No
Replicate Utility Changes(DREPLICATEUTI)	Whether Adabas utility change replication should be activated for a destination at Event Replicator Server startup or whether it should be activated for a specific target file. Valid values are "Yes" and "No". If "Yes" is specified, utility replication is activated; if "No" is specified, utility replication is not activated. For more information about replicating utility functions, read *Replicating Utility Functions*, in *Event Replicator for Adabas Concepts*.	No
Event Logging (*DEVENTLOG*)	Whether or not events should be logged by the Event Replicator Server and sent to this destination. This is the equivalent of specifying the *DEVENTLOG* directly in the Event Replicator Server startup job. Valid values are "Yes" or "No". When this optional parameter is set to "Yes", Event Replicator Server events are logged to the destination. When this parameter is set to "No" (the default), they are not. Event Replicator Server events are logged in URBS elements. In releases prior to 3.2.1, these URBS elements were sent to destinations related to the event itself. Starting with release 3.2.1, the URBS elements are also sent to any other destinations you have defined "Event Logging =Yes". If a related destination also is defined with "Event Logging =Yes", it will only receive one instance of the URBS element. To access this log of Event Replicator Server events in the destination queue, you must supply your own application that reads the event URBS elements in the destination queue. If such an application does not exist, the logged events simply sit in the queue.	No
Opened at Start (DOPEN parameter)	Whether or not the destination should be opened at Event Replicator Server startup. Valid values are "Yes", "No", or "Global", with "Global" as the default. When this parameter is set to "Yes", the destination is opened at Event Replicator Server startup. When this parameter is set to "No", the destination is *not* opened at Event Replicator Server startup. When this parameter is set to "Global", the decision to open the destination at Event Replicator Server startup depends on the setting of the **Global OPEN Value** (GOPEN) global parameter. If GOPEN=YES, the destination is opened at Event Replicator Server startup; if GOPEN=NO, it is not opened.	Global

Parameter Name	Specify	Default
	This is the equivalent of specifying the DOPEN parameter in the Event Replicator Server startup job.	

2 Supply TLOG parameter specifications for the File destination. The fields for these TLOG parameters can be found at the bottom of the **File Destination** panel:



Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Assign Level (DTLASSIGN)	The level of transaction logging that should occur when a transaction is assigned to a destination for output processing. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Completion Level (DTLCOMP)	The level of transaction logging that should occur when a transaction has been successfully output to the messaging system. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Write Level (DTLSLOGWRITE)	The level of transaction logging that should occur when a transaction has been successfully written to the SLOG file. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Read Level (DTLSLOGREAD)	The level of transaction logging that should occur when a transaction has been successfully read from the SLOG and is about to be queued for output to the destination. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging

Step 3. Save the File Destination Definition

To use Event Replicator Administration to save a File destination definition:

- When all specifications have been made to your satisfaction, click **OK** on the File destination panel.

  The File destination definition is saved in the Replicator system file.

Creating a Null Destination Definition

Using null destinations, data replication is tested without actually sending the data to any destination.

To create a null destination definition in Event Replicator Administration, complete the following steps:

Step 1. Access the Null Destination Definition Creation Area

To access the null destination definition creation area of Event Replicator Administration:

1. List the destination definitions in Event Replicator Administration, as described in ***Listing Destination Definitions***, elsewhere in this guide.

   The destination definitions are listed in detail-view.
2. Right-click on **Destinations** in the tree-view under **Replication Definitions**.

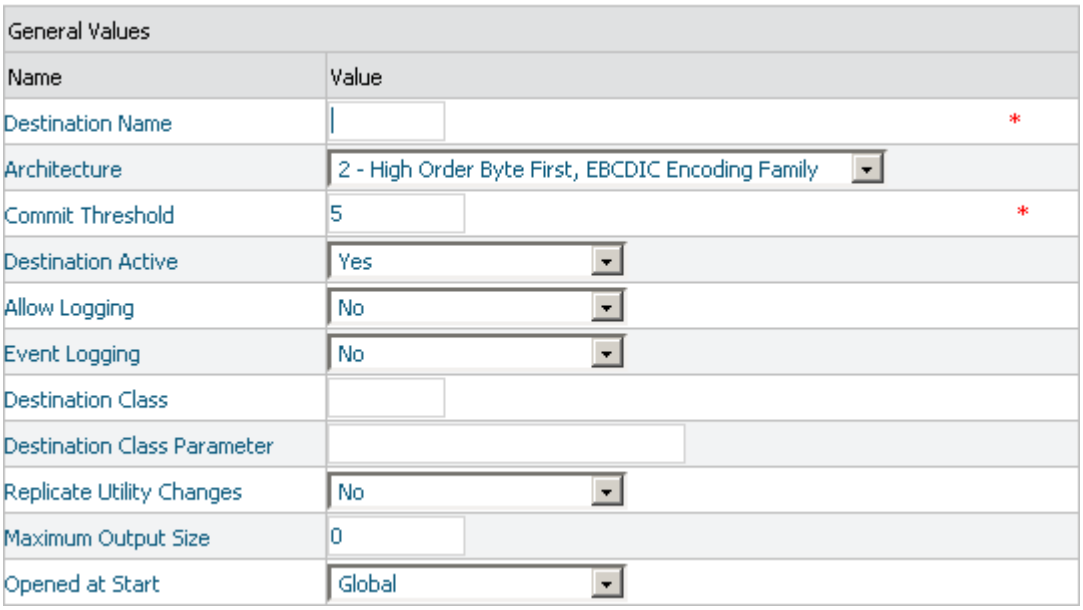
   A drop-down menu appears.
3. Click on **Create New NULL Destination** in the drop-down menu.

   A blank **NULL Destination** panel appears in detail-view.

Step 2. Specify General and TLOG Null Destination Parameters

To use Event Replicator Administration to supply general and TLOG specifications for a null destination definition, complete the following steps:

1 Supply general parameter specifications for the null destination. The fields for these general parameters can be found at the top of the **NULL Destination** panel:



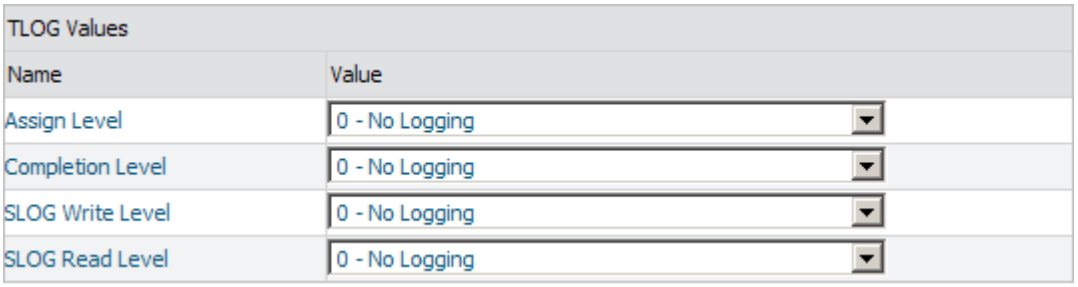
Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Destination Name (NAME)	The unique name for the null destination definition. The specified name must be alphanumeric and be between one and eight characters long.	---
Architecture (DARC)	The data architecture for fields in the URB* control structures sent to the null destination. For complete information on calculating a value for this parameter, read *DARC*, in *Event Replicator for Adabas Reference Guide* provided with your Event Replicator Administration documentation.	2 (High-order byte first, EBCDIC encoding)
Commit Threshold (DCOMMITTHRESHOLD)	The number of messages that will be sent to the null destination before a commit is performed for those messages. The term "commit" in this context means that the Event Replicator Server informs the messaging system that all messages sent (since the last commit) should be made permanent. In the case of	5

Parameter Name	Specify	Default
	WebSphere MQ, commit means that the Event Replicator Server will issue an MQCMIT call for the queue.	
Destination Active (DACTIVE)	Whether or not this destination definition should be activated for use once it is loaded by the Event Replicator Server. Valid values are "Yes" (load and activate the definition) or "No" (load, but do not activate the definition).	Yes
Allow Logging (DLOG)	Whether or not subscription logging should be activated for this destination definition. Valid values are "Yes" (activate subscription logging) or "No" (do not activate subscription logging).	No
Event Logging (*DEVENTLOG*)	Whether or not events should be logged by the Event Replicator Server and sent to this destination. This is the equivalent of specifying the *DEVENTLOG* directly in the Event Replicator Server startup job. Valid values are "Yes" or "No". When this optional parameter is set to "Yes", Event Replicator Server events are logged to the destination. When this parameter is set to "No" (the default), they are not. Event Replicator Server events are logged in URBS elements. In releases prior to 3.2.1, these URBS elements were sent to destinations related to the event itself. Starting with release 3.2.1, the URBS elements are also sent to any other destinations you have defined "Event Logging =Yes". If a related destination also is defined with "Event Logging =Yes", it will only receive one instance of the URBS element. To access this log of Event Replicator Server events in the destination queue, you must supply your own application that reads the event URBS elements in the destination queue. If such an application does not exist, the logged events simply sit in the queue.	No
Destination Class (DCLASS)	For the moment, leave this field blank. It is described later in this section.	---
Destination Class Parameter (DCLASSPARM)	For the moment, leave this field blank. It is described later in this section.	---
Replicate Utility Changes (DREPLICATEUTI)	Whether Adabas utility change replication should be activated for a destination at Event Replicator Server startup. Valid values are "Yes" and "No". If "Yes" is specified, utility replication is activated for the destination at Event Replicator Server startup; if "No" is specified, utility replication is not activated for the destination. For more information about replicating utility functions, read *Replicating Utility Functions*, in *Event Replicator for Adabas Concepts*.	No

Parameter Name	Specify	Default
Maximum Output Size (DMAXOUTPUTSIZE)	The maximum output size (in bytes) for the destination. This is the equivalent of specify the DMAXOUTPUTSIZE parameter directly in the Event Replicator Server startup job. Valid values are 0 or any integer ranging from 4096 through 2,147,483,647. You can specify the value for this parameter in a purely numeric form or use K at the end of the number to specify kilobytes. For example, DMAXOUTPUTSIZE=4K is the same as DMAXOUTPUTSIZE=4096. The value for this parameter will be used if it is less than or equal to the maximum output size for the Event Replicator Server (specified using the MAXOUTPUTSIZE global parameter) and less than or equal to the maximum output allowed for the messaging system queue being defined. If this value is larger than the MAXOUTPUTSIZE specification or the maximum output size allowed by the messaging system, the smaller value will be used. A value of 0 indicates that no specific limit is set for this destination. Instead, the smaller of the MAXOUTPUTSIZE specification or the maximum output size allowed by the messaging system will be used.	0
Opened at Start (DOPEN parameter)	Whether or not the destination should be opened at Event Replicator Server startup. Valid values are "Yes", "No", or "Global", with "Global" as the default. When this parameter is set to "Yes", the destination is opened at Event Replicator Server startup. When this parameter is set to "No", the destination is *not* opened at Event Replicator Server startup. When this parameter is set to "Global", the decision to open the destination at Event Replicator Server startup depends on the setting of the **Global OPEN Value** (GOPEN) global parameter. If GOPEN=YES, the destination is opened at Event Replicator Server startup; if GOPEN=NO, it is not opened. This is the equivalent of specifying the DOPEN parameter in the Event Replicator Server startup job.	Global

2 Supply TLOG parameter specifications for the null destination. The fields for these TLOG parameters can be found at the bottom of the **NULL Destination** panel:



Supply values for the these fields, as described in the following table:

Parameter Name	Specify	Default
Assign Level (DTLASSIGN)	The level of transaction logging that should occur when a transaction is assigned to a destination for output processing. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
Completion Level (DTLCOMP)	The level of transaction logging that should occur when a transaction has been successfully output to the messaging system. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Write Level (DTLSLOGWRITE)	The level of transaction logging that should occur when a transaction has been successfully written to the SLOG file. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging
SLOG Read Level (DTLSLOGREAD)	The level of transaction logging that should occur when a transaction has been successfully read from the SLOG and is about to be queued for output to the destination. Valid values are "no logging", "log event and output transaction data", "log event, output transaction, and file/record data", or "log event and all available output information".	no logging

Step 3. (Optional) Specify Destination Class Information, If Applicable

The Destination Class and Destination Class Parameter fields can be used to invoke and pass parameters to the Event Replicator Target Adapter for this destination. The fields are located in the middle of the general fields for the null destination. Do not use these fields unless you want to invoke and pass parameters to the Event Replicator Target Adapter for the destination or unless otherwise requested by a Software AG support representative.

If applicable, use the Destination Class (DCLASS) field to specify the destination class for this destination definition. Valid values are "NONE" (interpreted as a blank) or "SAGTARG" (if Event Replicator Target Adapter processing should be invoked). There is no default.

If you specify a value for the Destination Class field, you can optionally use the Destination Class Parameter (DCLASSPARM) field to specify up to 120 bytes of character data to be passed to the optional destination output user exit.

If DCLASS=SAGTARG is specified (if the Destination Class field is set to "SAGTARG") to invoke the Event Replicator Target Adapter for a destination, you may want to specify one or more of the following keyword parameters:

Note: These parameter keywords must be specified in uppercase.

NOSPRE

Specify the "NOSPRE" keyword in the DCLASSPARM parameter if you do not want the subscription name to prefix the names of the tables produced by the Event Replicator Target Adapter. When "NOSPRE" is specified, the schema file name (Predict view name) alone is used for the table names; when "NOSPRE" is *not* specified, the subscription name prefixes the schema file name in the table names.

Note: Oracle identifiers are limited to 30 characters. If NOSPRE is *not* specified and an Oracle RDBMS is used by the Event Replicator Target Adapter, the identifier names may exceed 30 characters and errors may occur. We recommend using NOSPRE if an Oracle RDBMS is also used.

OPTIONS

The OPTIONS keyword parameter can be used to specify options for the destination. Specify one or more of the options described in the following table, using the syntax OPTIONS=*val1*, *val2*,...

Option Value	Description
1	This option is no longer supported. If specified, it will be ignored.
2	Specify OPTIONS=2 to indicate that long names should be used. This option will cause long names to be sent in place of the default short names used for various elements and attributes. Short names are the default and save on the amount of data being transferred. Long names make for better readability. For example, the short name <F> would appear as <Field> using long names.
4	Specify OPTIONS=4 to ensure that invalid XML characters found in alphanumeric fields are not translated to spaces.
8	Specify OPTIONS=8 to ensure that trailing blanks in alphanumeric fields are not removed.
16	Specify OPTIONS=16 to ensure that characters used by XML are not replaced automatically with predefined entity references. For example, if OPTIONS=16 is set, the ampersand (&) character would not be replaced with the literal "&".
32	Specify OPTIONS=32 to send the full image on an update. The full before image (if available) and after image of all fields are sent for an update, even if the field values were not changed or are null.

TRACE

The TRACE keyword parameter can be used to specify the contents of the trace. Specify the TRACE keyword parameter using the syntax `TRACE=nnnn`. Possible values of the TRACE (*nnnn*) are listed in the following table. However, if you want to trace multiple control blocks, add their trace values together and enter the total value. For example, to trace the before and after images of the URBD control blocks, you would specify `TRACE=24` because the sum of 8 (URBD control block before image) and 16 (URBD control block after image) is 24.

Note: If tracing is enabled using this keyword parameter, be sure to include the following JCL statement in the startup JCL of the Event Replicator Server:

```
//DDTRACE1 DD SYSOUT=X
```

Trace Value	Description
1	Trace the URBS control block.
2	Trace the URBT control block.
4	Trace the URBR control block.
8	Trace the URBD control block before image.
16	Trace the URBD control block after image.
32	Trace the URBF/URBG control block before image.
64	Trace the URBF/URBG control block after image.
128	Trace the send buffer.
256	Trace the URBY control block.
512	Trace the URBO control block.
1024	Trace the output parameters.
2048	Trace the subscription table.

Step 4. Save the Null Destination Definition

To use Event Replicator Administration to save a null destination definition:

- When all specifications have been made to your satisfaction, click **OK** on the **NULL Destination** panel.

  The null destination definition is saved in the Replicator system file.

Modifying Destination Definitions

To use Event Replicator Administration to modify a destination definition in the Replicator system file:

1 List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*, elsewhere in this guide.

The destination definitions are listed in detail-view.

2 Locate the definition you want to modify in the table in detail-view and click on it.

The Destination Definition panel appears in detail-view listing the current settings for the destination definition you selected.

3 Click the **Modify** button.

The destination parameters you can modify become editable in detail-view. For information on modifying these parameters, read the description of adding that type of destination definition, elsewhere in this section.

4 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the changes.

Copying Destination Definitions

To use Event Replicator Administration to copy a destination definition in the Replicator system file:

1 List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*, elsewhere in this guide.

The destination definitions are listed in detail-view.

2 Locate the definition you want to copy in the table in detail-view and click on it.

The Destination Definition panel appears in detail-view listing the current settings for the destination definition you selected.

3 Click the **Copy** button.

A copy of the destination definition is created and its parameter values appear in detail-view.

4 Specify a new, unique name for the copy of the destination definition in the Value column for the **Destination Name** parameter.

5 If you wish, modify any other parameters for the new copy in detail-view. For information on modifying the parameters, read the description of adding that type of destination definition, elsewhere in this section.

6 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the copy.

Activating and Deactivating Destination Definitions

You can use Event Replicator Administration to activate and deactivate destination definitions.

Caution: Be careful when you activate and deactivate replication definitions and databases, especially if replication is ongoing at the time. Whenever you activate or deactivate definitions or databases, you run the risk of altering what data is replicated and how that replication occurs. If the Event Replicator Server receives data from an Adabas database for which it has no active definitions, replication simply does not occur.

This section covers the following topics:

Activating Destination Definitions

To use Event Replicator Administration to activate a destination definition:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Destinations** in the tree-view under **Replication**.

A table listing the destination definitions in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the definition you want to activate.

Details about the destination appear in detail-view.

6 Click the **Activate** button.

Note: This button will not be available if the definition is already activated.

The destination definition is activated.

Deactivating Destination Definitions

To use Event Replicator Administration to deactivate a destination definition:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Destinations** in the tree-view under **Replication**.

A table listing the destination definitions in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the definition you want to deactivate.

Details about the destination appear in detail-view.

6 Click the **Deactivate** button.

Note: This button will not be available if the definition is already deactivated.

The destination definition is deactivated.

Opening and Closing Destinations

You can use Event Replicator Administration to open and close destinations. This section covers the following topics:

- Opening Destinations
- Closing Destinations

Opening Destinations

To use Event Replicator Administration to open a destination:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Destinations** in the tree-view under **Replication**.

A table listing the destination definitions in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the definition you want to open.

Details about the destination appear in detail-view.

6 Click the **Open** button.

Note: This button will not be available if the destination is already opened.

The destination is opened.

Closing Destinations

To use Event Replicator Administration to close a destination:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Destinations** in the tree-view under **Replication**.

A table listing the destination definitions in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the definition you want to close.

Details about the destination appear in detail-view.

6 Click the **Close** button.

Note: This button will not be available if the destination is already closed.

The destination is closed.

Deleting Destination Definitions

To use Event Replicator Administration to delete a destination definition in the Replicator system file:

1 List the destination definitions in Event Replicator Administration, as described in *Listing Destination Definitions*, elsewhere in this guide.

The destination definitions are listed in detail-view.

2 Locate the definition you want to delete in the table in detail-view and click on it.

The Destination Definition panel appears in detail-view listing the current settings for the destination definition you selected.

3 Click the **Delete** button.

A confirmation panel appears verifying that you want to delete the definition. If you click **Yes** (indicating that you do want to delete the definition), the definition is deleted. If you click **No** (indicating that you do not want to delete the definition), the definition is not deleted.

7 Maintaining Input Queue (IQUEUE) Definitions

An IQUEUE definition defines the input queue on which Event Replicator for Adabas should listen for requests from webMethods EntireX and WebSphere MQ targets. At least one definition is required for every webMethods EntireX or WebSphere MQ target you intend to use.

You can create webMethods EntireX and WebSphere MQ input queue definitions using Event Replicator Administration.

Listing IQUEUE Definitions

To use Event Replicator Administration to list the IQUEUE definitions stored in the Replicator system file:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Replication Definitions** in tree-view under the selected database.

3 Click on **Input Queues** in the tree-view under **Replication Definitions**.

A table listing the IQUEUE definitions in the Replicator system file appears in detail-view.

Creating a webMethods EntireX IQUEUE Definition

Prior to using webMethods EntireX as the messaging subsystem, be sure to read *Using webMethods EntireX as the Messaging System*, in *Event Replicator for Adabas Administration and Operations Guide* provided with your Event Replicator Administration documentation.

To use Event Replicator Administration to add a webMethods EntireX IQUEUE definition to the Replicator system file:

1 List the IQUEUE definitions in Event Replicator Administration, as described in *Listing IQUEUE Definitions*, elsewhere in this guide.

The IQUEUE definitions are listed in detail-view.

2 Right-click on **Input Queues** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3 Click on **Create New EntireX Broker Input Queue Definition** in the drop-down menu.

A blank **EntireX Broker Input Queue** panel appears in detail-view.

4 Fill in the fields in Value columns of the detail-view table as described below:

Parameter Name	Specify	Default
Input Queue Name (IQUEUE NAME)	A unique name for the IQUEUE definition. The name must use alphanumeric characters and be between one and 8 characters long.	---
Broker ID (IQETBBROKERID)	The webMethods EntireX Broker ID you want used for this IQUEUE definition. For the valid values of this parameter, check with your webMethods EntireX Broker development staff and the webMethods EntireX Broker documentation. A maximum of 32 characters can be specified. Broker IDs come in two formats: one for TCP/IP communications and one for Adabas SVC communication. For TCP/IP communications, the format is: *ip-address*:*port-number*:TCP In this case, the *ip-address* setting is the TCP/IP IP address and the *port-number* setting should match the webMethods EntireX PORT parameter. For Adabas SVC communications, the format is: '*broker-id*:SVC*nnn*:NET' In this case, the *broker-id* setting should match the webMethods EntireX BROKER-ID parameter in the Broker ETBFILE DD. The *nnn* setting should match either the webMethods EntireX ADASVC or ADA5SVC parameters in the Broker PARMS DD statement.	ETBBROKERID value
Broker Service (IQETBSERVICE)	The webMethods EntireX service you want used for this IQUEUE definition. This should be the same as the value specified for the SERVICE parameter in webMethods EntireX. The ID can be up to 32 characters long. For the valid values of this parameter, check with your webMethods EntireX Broker development staff and the webMethods EntireX Broker documentation.	---
Service Name (IQETBSERVICENAME)	The name of the webMethods EntireX service you want used for this IQUEUE definition. This should be the same as the value specified for the SERVER parameter in webMethods EntireX. The name can be up to 32 characters long. For the valid values of this parameter, check with your webMethods EntireX Broker development staff and the webMethods EntireX Broker documentation.	---
Service Class (IQETBSERVICECLASS)	The webMethods EntireX service class you want used for this IQUEUE definition. This should be the same as the value specified for the CLASS parameter in webMethods EntireX. The name can be up to 32 characters long. For the valid values of this parameter, check with your webMethods EntireX Broker	---

Parameter Name	Specify	Default
	development staff and the webMethods EntireX Broker documentation.	
Input Queue Buffer Length (IQBUFLEN parameter)	The length, in bytes, of the input buffer associated with this input queue. Valid values range from "2048" through "2,147,483,647". However, the practical maximum value is restricted by the amount of virtual storage available in the Event Replicator Server address space. If you enter a value less than "2048", "2048" is used. This is the equivalent of specifying the IQBUFLEN parameter in the Event Replicator Server startup job. This value should be set to a value greater than or equal to the largest message that will be received by the input queue. When the input queue will receive data as a part of node-to-node replication, the largest message will be limited by the minimum of the following Event Replicator Server settings: MAXOUTPUTSIZE parameter, DMAXOUTPUTSIZE parameter (if specified for the destination), or the message limit imposed by the messaging system.	2048
Open Retry Count(IQRETRYCOUNT parameter)	The number of times that an attempt to open the input queue will be retried at the interval specified by the **Retry Interval** parameter. This is the equivalent of specifying the IQRETRYCOUNT parameter directly in the Event Replicator Server startup job. Valid values range from 0 through 2,147,483,647 or the literal "GLOBAL". If the value "GLOBAL" is specified for this parameter, the specification for the **Retry Count global variable** will be used. Any retry attempts will occur at the interval specified by the **Retry Interval** parameter. A value of zero indicates that no retry attempt to open this input queue should occur.	The value of the **Retry Count global variable**
Open Retry Interval (IQRETRYINTERVAL parameter)	The default number of seconds between retry attempts to open the input queue identified by this definition. This is the equivalent of specifying the IQRETRYINTERVAL parameter directly in the Event Replicator Server startup job. Valid values are 0, 5 through 2,147,483,647, or the literal "GLOBAL". If the value "GLOBAL" is specified for this parameter, the specification for the **Retry Interval global variable** will be used. A value of zero indicates that no retry attempt to open this input queue should occur. Except for a specification of zero, the minimum value that can be specified for this parameter is 5 seconds.	The value of the **Retry Interval global variable**

5 When all specifications have been made to your satisfaction, click **OK** to save the IQUEUE definition.

Creating a WebSphere MQ IQUEUE Definition

Prior to using WebSphere MQ as the messaging subsystem, be sure to read *Using WebSphere MQ as the Messaging System*, in *Event Replicator for Adabas Administration and Operations Guide* provided with your Event Replicator Administration documentation.

To use Event Replicator Administration to add a WebSphere MQ IQUEUE definition using the Replicator system file:

1 List the IQUEUE definitions in Event Replicator Administration, as described in ***Listing IQUEUE Definitions***, elsewhere in this guide.

The IQUEUE definitions are listed in detail-view.

2 Right-click on **Input Queues** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3 Click on **Create New MQSeries Input Queue Definition** in the drop-down menu.

A blank **MQSeries Input Queue** panel appears in detail-view.

4 Fill in the fields in Value columns of the detail-view table as described below:

Parameter Name	Specify	Default
Input Queue Name (`IQUEUE NAME`)	A unique name for the IQUEUE definition. The name must use alphanumeric characters and be between one and 8 characters long.	---
Queue Manager Name (`IQMQQMGRNAME`)	The name of the WebSphere MQ queue manager you want used for this IQUEUE definition. The name can be up to 48 characters long. For the valid values of this parameter, check with your WebSphere MQ development staff and the WebSphere MQ documentation.	---
MQ Queue name (`IQMQQNAME`)	The name of the WebSphere MQ queue you want used for this IQUEUE definition. The name can be up to 48 characters long. For the valid values of this parameter, check with your WebSphere MQ development staff and the WebSphere MQ documentation.	---
Open Retry Count(`IQRETRYCOUNT` parameter)	The number of times that an attempt to open the input queue will be retried at the interval specified by the **Retry Interval** parameter. This is the equivalent of specifying the	The value of the **Retry Count global variable**

Parameter Name	Specify	Default
	IQRETRYCOUNT parameter directly in the Event Replicator Server startup job. Valid values range from 0 through 2,147,483,647 or the literal "GLOBAL". If the value "GLOBAL" is specified for this parameter, the specification for the **Retry Count global variable** will be used. Any retry attempts will occur at the interval specified by the **Retry Interval** parameter. A value of zero indicates that no retry attempt to open this input queue should occur.	
Open Retry Interval (IQRETRYINTERVAL parameter)	The default number of seconds between retry attempts to open the input queue identified by this definition. This is the equivalent of specifying the IQRETRYINTERVAL parameter directly in the Event Replicator Server startup job. Valid values are 0, 5 through 2,147,483,647, or the literal "GLOBAL". If the value "GLOBAL" is specified for this parameter, the specification for the **Retry Interval global variable** will be used. A value of zero indicates that no retry attempt to open this input queue should occur. Except for a specification of zero, the minimum value that can be specified for this parameter is 5 seconds.	The value of the **Retry Interval global variable**

5 When all specifications have been made to your satisfaction, click **OK** to save the IQUEUE definition.

Modifying IQUEUE Definitions

To use Event Replicator Administration to modify an IQUEUE definition in the Replicator system file:

1 List the IQUEUE definitions in Event Replicator Administration, as described in *Listing IQUEUE Definitions*, elsewhere in this guide.

The IQUEUE definitions are listed in detail-view.

2 Locate the definition you want to modify in the table in detail-view and click on it.

The appropriate IQUEUE definition panel appears in detail-view listing the current settings for the IQUEUE definition you selected.

3 Click the **Modify** button.

The IQUEUE parameters you can modify become editable in detail-view. For information on modifying these parameters, read the description of the creation of that type IQUEUE definition, elsewhere in this section.

4 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the changes.

Copying IQUEUE Definitions

To use Event Replicator Administration to copy an IQUEUE definition in the Replicator system file:

1 List the IQUEUE definitions in Event Replicator Administration, as described in ***Listing IQUEUE Definitions***, elsewhere in this guide.

The IQUEUE definitions are listed in detail-view.

2 Locate the definition you want to copy in the table in detail-view and click on it.

The appropriate IQUEUE definition panel appears in detail-view listing the current settings for the IQUEUE definition you selected.

3 Click the **Copy** button.

A copy of the IQUEUE definition is created and its parameter values appear in detail-view.

4 Specify a new, unique name for the copy of the IQUEUE definition in the Value column for the **Input Queue Name** parameter.

5 If you wish, modify any other parameters for the new copy in detail-view. For information on modifying the parameters, read the description of the creation of that type IQUEUE definition, elsewhere in this section.

6 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the copy.

Opening and Closing IQUEUE Definitions

You can use Event Replicator Administration to open and close IQUEUE definitions. This section covers the following topics:

- Opening IQUEUE Definitions

Opening IQUEUE Definitions

To use Event Replicator Administration to open an IQUEUE:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Input Queues** in the tree-view under **Replication**.

A table listing the IQUEUE definitions in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the definition you want to open.

Details about the IQUEUE definition appear in detail-view.

6 Click the **Open** button.

The IQUEUE is opened.

Closing IQUEUE Definitions

To use Event Replicator Administration to close an IQUEUE:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Input Queues** in the tree-view under **Replication**.

A table listing the IQUEUE definitions in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the definition you want to close.

Details about the IQUEUE definition appear in detail-view.

6 Click the **Close** button.

The IQUEUE is closed.

Deleting IQUEUE Definitions

To use Event Replicator Administration to delete an IQUEUE definition in the Replicator system file:

1 List the IQUEUE definitions in Event Replicator Administration, as described in ***Listing IQUEUE Definitions***, elsewhere in this guide.

The IQUEUE definitions are listed in detail-view.

2 Locate the definition you want to delete in the table in detail-view and click on it.

The appropriate IQUEUE definition panel appears in detail-view listing the current settings for the IQUEUE definition you selected.

3 Click the **Delete** button.

A confirmation panel appears verifying that you want to delete the definition. If you click **Yes** (indicating that you do want to delete the definition), the definition is deleted. If you click **No** (indicating that you do not want to delete the definition), the definition is not deleted.

8 Maintaining Resend Buffer Definitions

A resend buffer definition defines a resend buffer that can be used by any subscription to expedite the retransmission of a transaction. No resend buffer definitions are required. Resend buffers are defined using the RESENDBUFFER initialization parameter, but should be maintained using Event Replicator Administration.

Listing Resend Buffer Definitions

To use Event Replicator Administration to list the resend buffer definitions stored in the Replicator system file:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Replication Definitions** in tree-view under the selected database.

3 Click on **Resend Buffers** in the tree-view under **Replication Definitions**.

A table listing the resend buffer definitions in the Replicator system file appears in detail-view.

Adding Resend Buffer Definitions

To use Event Replicator Administration to add a resend buffer definition to the Replicator system file:

1 List the resend buffer definitions in Event Replicator Administration, as described in *Listing Resend Buffer Definitions*, elsewhere in this guide.

The resend buffer definitions are listed in detail-view.

2 Right-click on **Resend Buffers** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3 Click on **Create New Resend Buffer** in the drop-down menu.

A blank **Resend Buffer** panel appears in detail-view.

4 Fill in the fields in Value columns of the detail-view table as described below:

Parameter Name	Description	Default
Resend Buffer Name (RESENDBUFFERNAME)	Specify a unique name for the resend buffer definition. The name must use alphanumeric characters and be between one and 8 characters long. There are some constraints on the name. It must: ■ Be comprised of one to eight uppercase, alphanumeric characters and can include the special characters "@", "$", or "#". ■ Not begin with a numeric character or a blank. ■ Have no embedded blanks. ■ Be padded on the right side with blanks if the name is less than eight characters. ■ Not begin with the letters "SYS".	---
Resend Buffer Size (RSIZE)	Specify the amount of storage allocated to the buffer. The value must be numeric and must be specified in KB units. The default is 32 (32K), and the maximum is 2,097,151 (2,097,151K). If a resend buffer is defined for a subscription that delivers data to multiple destinations, multiple copies of the sent data may be saved, one copy for each destination. The specification of **Resend Buffer Size** must be large enough to accommodate these multiple copies of the data.	32

5 When all specifications have been made to your satisfaction, click **OK** to save the resend buffer definition.

Modifying Resend Buffer Definitions

To use Event Replicator Administration to modify a resend buffer definition in the Replicator system file:

1 List the resend buffer definitions in Event Replicator Administration, as described in *Listing Resend Buffer Definitions*, elsewhere in this guide.

The resend buffer definitions are listed in detail-view.

2 Locate the definition you want to modify in the table in detail-view and click on it.

The **Resend Buffer** panel appears in detail-view listing the current settings for the resend buffer definition you selected.

3 Click the **Modify** button.

The resend buffer parameters you can modify become editable in detail-view. For information on modifying these parameters, read the description of *Adding Resend Buffer Definitions*, elsewhere in this section.

4 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the changes.

Copying Resend Buffer Definitions

To use Event Replicator Administration to copy a resend buffer definition in the Replicator system file:

1 List the resend buffer definitions in Event Replicator Administration, as described in *Listing Resend Buffer Definitions*, elsewhere in this guide.

The resend buffer definitions are listed in detail-view.

2 Locate the definition you want to copy in the table in detail-view and click on it.

The **Resend Buffer** panel appears in detail-view listing the current settings for the resend buffer definition you selected.

3 Click the **Copy** button.

A copy of the resend buffer definition is created and its parameter values appear in detail-view.

4 Specify a new, unique name for the copy of the resend buffer definition in the Value column for the **Resend Buffer Name** parameter.

5 If you wish, modify any other parameters for the new copy in detail-view. For information on modifying the parameters, read the description of *Adding Resend Buffer Definitions*, elsewhere in this section.

6 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the copy.

Deleting Resend Buffer Definitions

To use Event Replicator Administration to delete a resend buffer definition in the Replicator system file:

1 List the resend buffer definitions in Event Replicator Administration, as described in *Listing Resend Buffer Definitions*, elsewhere in this guide.

The resend buffer definitions are listed in detail-view.

2 Locate the definition you want to delete in the table in detail-view and click on it.

The **Resend Buffer** panel appears in detail-view listing the current settings for the resend buffer definition you selected.

3 Click the **Delete** button.

A confirmation panel appears verifying that you want to delete the definition. If you click **Yes** (indicating that you do want to delete the definition), the definition is deleted. If you click **No** (indicating that you do not want to delete the definition), the definition is not deleted.

9 Maintaining Transaction Filter Definitions

A transaction filter definition specifies filter conditions for replication, based on the values of fields in the database records. No transaction filter definitions are required. Transaction filter definitions are defined using the `FILTER` initialization parameter, but should be maintained using Event Replicator Administration.

Listing Transaction Filter Definitions

To use Event Replicator Administration to list the transaction filter definitions stored in the Replicator system file:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Replication Definitions** in tree-view under the selected database.

3 Click on **Transaction Filters** in the tree-view under **Replication Definitions**.

A table listing the transaction filter definitions in the Replicator system file appears in detail-view.

Adding Transaction Filter Definitions

To add transaction filter definitions, complete the following steps:

- Step 1. Access the Transaction Filter Definition Area of Event Replicator Administration
- Step 2. Specify a Transaction Filter Definition Name and Type
- Step 3. Add Filter Conditions to the Transaction Filter Definition
- Step 4. Save the Transaction Filter Definition

Step 1. Access the Transaction Filter Definition Area of Event Replicator Administration

To access the transaction filter definition area, complete the following steps:

1 List the transaction filter definitions in Event Replicator Administration, as described in *Listing Transaction Filter Definitions*, elsewhere in this guide.

The transaction filter definitions are listed in detail-view.

2 Right-click on **Transaction Filters** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3 Click on **Create New Internal Transaction Filter** in the drop-down menu.

A blank **Internal Transaction Filter** panel appears in detail-view.

Step 2. Specify a Transaction Filter Definition Name and Type

To specify a transaction filter definition name and type:

1 On the **Internal Transaction Filter** panel, supply values for the following fields:

Field	Description	Default
Transaction Filter Name (`FILTER NAME`)	Specify a unique name for the transaction filter definition. The name must use alphanumeric characters and be between one and 8 characters long.	---
Exclude or Include Records (`FRECORDS`)	Specify "Include" to include (replicate) the records selected by the filter definition or "Exclude" to exclude (do not replicate) records selected by the filter definition.	Include
Filter Conditions	See the next step in these instructions.	---

2 Click **OK**.

The internal transaction filter is created. If you added filter conditions, prior to click OK, the filter conditions are included in the transaction filter definition. Filter conditions can be added and maintained using the Event Replicator Administration, as described in the next step in these instructions.

Step 3. Add Filter Conditions to the Transaction Filter Definition

For more information about rules of filter conditions, read *Rules for Writing Filter Conditions*, elsewhere in this guide.

To add filter conditions to the transaction filter definition:

1 Click on the checkmark in the **Filter Conditions** field on the **Internal Transaction Filter** panel to display a list of filter conditions.

Note: You might need to click the **Modify** button on the **Internal Transaction Filter** screen to make the **Filter Conditions** checkmark appear.

The **Filter Conditions** table appears in detail-view.

2 Add, modify, and delete the filter condition definitions in the **Filter Conditions** table.

- To add a new filter condition definition, click **New**. A new, blank, line appears in the **Filter Conditions** table. Update the fields in the filter table for the new filter definition, as described later in this step.

- To modify a filter condition definition, edit the fields in the table as described later in this step.

  Note: The filter condition definition name cannot be changed.

- To delete a filter condition definition, select the definition you want to delete by clicking the check box for the definition in the **All** column of the **Filter Conditions** table. More than one filter condition definition can be selected at a time. To select all filter conditions, click on the check box in the **All** column heading. Once all the filter condition definitions you want to delete are selected, click **Delete**.

In the **Value** field of the **Filter Values** table, specify a value against which the source field will be compared. Only one value can be specified in each Value field. Up to 128 Value fields are available in which you can specify values; use the scroll bar to scroll through them.

The fields in a filter condition definition are described in the following table:

Parameter Name	Specify	Default
All	A selection field you can use to select one or more filter conditions in the **Filter Conditions** table.	none selected.
Group	A group number you can use to group field filters together within a transaction filter definition. All of the field filters with the same group number are blocked and logically ANDed together when the filters are examined during subscription processing. In other words, a field in the database must meet all of the criteria of the group before it is selected. Likewise, different groups of field filters are logically ORed together. In other words, a field need only meet the criteria specified by one of the groups to be selected. Valid values are numbers ranging from "1" through "999". The equivalent processing in the Event Replicator Server startup job is available through the use of the OR keyword within a series of `FFIELD` parameters.	---
Source Field (`FFIELD`)	The two-byte Adabas field code for the field to be compared. This field must be in the format buffer specified for the file in the SFILE definition of the subscription.	---
Source PE (`FSPE`)	The index number (occurrence) of the periodic group (PE) to which the condition relates if the source field in this field filter is a PE field. Valid values range from 0 through 191.	0, indicating the source field is not a PE field.
Source MU (`FSMU`)	The index number of the multiple-value field (MU) to which the condition relates if the source field in this field filter is an MU field. Valid values range from 0 through 191.	0, indicating the source field is not an MU field.

Parameter Name	Specify	Default
Source Image (FSIMAGE)	Whether the source field is in the after image, before image, or the default image of the record. Valid values are "After", "Before" and "None".	"After" for adds and updates; "Before" for deletes
Source Begin (FSBEGIN)	The starting byte number of the partial Adabas source field to be compared. This field is only used if you want to specify a partial field for comparison and only if the field is of alphanumeric or binary format. **Note:** The format of the complete field is used for partial field comparisons. Valid comparisons of different field types are listed in ***Field Type Considerations***, elsewhere in this guide. For fixed length fields, valid values range from "1" (the start of the field) through the maximum length of the field (the last byte of the field). For variable length fields, valid values range from "1" (the start of the field) to the maximum length allowed for that field type. Counting occurs from left to right beginning with 1 for fields defined with alphanumeric format, and from right to left beginning with 1 for fields defined with binary format.	1
Source Length (FSLENGTH)	The numeric length of the partial Adabas source field that should be compared. This field is only used if you want to specify a partial field for comparison and only if the field is of alphanumeric or binary format. **Note:** The format of the complete field is used for partial field comparisons. Valid comparisons of different field types are listed in ***Field Type Considerations***, elsewhere in this guide. For fixed length fields, errors will occur if the sum of the values of the Source Begin and Source Length parameters exceeds the fixed length of the field. For variable length fields, the sum of the values of the Source Begin and Source Length parameters must not exceed the maximum length of the field plus 1. For example, if a variable length field has format "A" with a maximum length of 253 bytes, settings of Source Begin=1 and Source Length=253 are valid, but settings of Source Begin=2 and Source Length=254 are not.	If Source Begin is not specified, the default value is the entire field. If Source Begin is specified, the default value is the maximum length of the field minus the value of the Source Begin parameter plus 1.
Condition (FCOND)	A condition operator code for the filter. Valid values are "equal to", "not equal to", "less than", "less than or equal to", "greater than", or "greater than or equal to" . When "equal to" or "not equal to" are specified, multiple target values and target values using wildcards can be tested. For all other condition codes, only single target values without wildcards can be tested.	equal to

Parameter Name	Specify	Default
Target Field (FTARGET)	The two-byte Adabas field code for the field with which the source field will be compared. This field must be in the same record as the source field. This field is mutually exclusive with the Target Value *n* fields. If you specify the target field code, you cannot specify values in the Target Value *n* fields.	---
Target PE (FTPE)	The index number of the periodic group (PE) to which the condition relates if the target field in this field filter is a PE field. Valid values range from 0 through 191. This field is mutually exclusive with the Target Value *n* fields. If you specify the target field code, you cannot specify values in the Target Value *n* fields.	0, indicating the target field is not a PE field.
Target MU (FTMU)	The index number (occurrence) of the multiple-value field (MU) to which the condition relates if the target field in this field filter is an MU field. Valid values range from 0 through 191. This field is mutually exclusive with the Target Value *n* fields. If you specify the target field code, you cannot specify values in the Target Value *n* fields.	0, indicating the target field is not an MU field.
Target Image (FTIMAGE)	Whether the target field is in the after image, before image, or the default image of the record. Valid values are "After", "Before" and "None". This field is mutually exclusive with the Target Value *n* fields. If you specify the target field code, you cannot specify values in the Target Value *n* fields.	"After" for adds and updates; "Before""" for deletes
Target Begin (FTBEGIN)	The starting byte number of the partial Adabas target field at which the comparison should begin. This field should only be specified if you want to specify a partial field for comparison, if the field is of alphanumeric or binary format, and only if an Adabas target field (Target Field) is specified. **Note:** The format of the complete field is used for partial field comparisons. Valid comparisons of different field types are listed in ***Field Type Considerations***, elsewhere in this guide. For fixed length fields, valid values range from "1" (the start of the field) through the maximum length of the field (the last byte of the field). For variable length fields, valid values range from "1" (the start of the field) to the maximum length allowed for that field type. Counting occurs from left to right beginning with 1 for fields defined with alphanumeric format, and from right to left beginning with 1 for fields defined with binary format.	1

Parameter Name	Specify	Default
Target Length (FTLENGTH)	The numeric length of the partial Adabas target field that should be used for the comparison. This field should only be specified if you want to specify a partial field for comparison, if the field is of alphanumeric or binary format, and only if an Adabas target field (Target Field) is specified. **Note:** The format of the complete field is used for partial field comparisons. Valid comparisons of different field types are listed in *Field Type Considerations*, elsewhere in this guide. For fixed length fields, errors will occur if the sum of the values of the Target Begin and Target Length parameters exceeds the fixed length of the field. For variable length fields, the sum of the values of the Target Begin and Target Length parameters must not exceed the maximum length of the field plus 1. For example, if a variable length field has format "A" with a maximum length of 253 bytes, settings of Target Begin=1 and Target Length=253 are valid, but settings of Target Begin=2 and Target Length=254 are not.	If Target Begin is not specified, the default value is the entire field. If Target Begin is specified, the default value is the maximum length of the field minus the value of the Target Begin parameter plus 1.
Target Value (FLIST)	See the next step in these instructions. Click on the checkmark in this field to bring up the **Filter Values** panel in detail-view. You can use the **Filter Values** panel to specify one or more values for comparison in the filter condition. This field is mutually exclusive with the Target Field, Target Image, Target MU, and Target PE fields. You cannot specify values for the Target Field, Target Image, Target MU, or Target PE fields if you have specified a value for this field.	---

3 If you elected to include filter values in a filter condition and clicked the checkmark in the **Target Value** field on the **Filter Conditions** screen, the **Filter Values** panel appears in detail-view. You can use this panel to specify one or more values for the filter condition.

- To add a new filter value, click **New**. A new, blank, line appears in the **Filter Values** table. Update the fields in the filter values table for the filter definition, as described later in this step.
- To modify a filter value, edit the fields in the table as described later in this step.
- To delete a filter value, select the definition you want to delete by clicking the check box for the definition in the **All** column of the **Filter Conditions** table. More than one filter condition definition can be selected at a time. To select all filter conditions, click on the check box in the **All** column heading. Once all the filter condition definitions you want to delete are selected, click **Delete**.

The fields in a filter condition definition are described in the following table:

Parameter Name	Specify	Default
All	A selection field you can use to select one or more filter values in the **Filter Values** table.	none selected.

Value (FLIST)	This field is mutually exclusive with the Target Field, Target Image, Target MU, and Target PE fields. You cannot specify values for the Target Field, Target Image, Target MU, or Target PE fields if you have specified a value for this field.
	
	The following information should be considered when maintaining filter condition values:
	
	■ Strings that include blanks should be enclosed in single quotes. Apostrophes in strings must be doubled (for example: 'six o"clock'). A maximum of 254 characters can be specified for each value.
	■ Each value may consist of either free-format characters or a mix of elements specified using the A() or X() notation.
	  ■ If free-format data consists entirely of numeric data (including an optional leading "+" or "-" character) it is treated as a numeric value.
	  ■ If a value (or part of a value) is specified using A() notation, it will be treated as alphabetic data.
	  ■ Hexadecimal values may be specified using X() notation.
	■ A value must be specified entirely as free-format data, or composed of one or more A() or X() subelements. If a value begins with an A() or X() subelement all remaining subelements of the value must be so specified.
		--- 	

When all filter values have been made to your satisfaction, click **OK** to save them.

The **Filter Conditions** panel appears in detail-view.

4 Repeat Steps 2 and 3 until all filter conditions and any necessary filter values have been specified. When all specifications have been made to your satisfaction, click **OK** to save the transaction filter conditions.

Step 4. Save the Transaction Filter Definition

To use Event Replicator Administration to save a transaction filter definition:

- When all specifications have been made to your satisfaction, click **OK** to save the definition in the Replicator system file.

Modifying Transaction Filter Definitions

To use Event Replicator Administration to modify a transaction filter definition in the Replicator system file:

1 List the transaction filter definitions in Event Replicator Administration, as described in *Listing Transaction Filter Definitions*, elsewhere in this guide.

The transaction filter definitions are listed in detail-view.

2 Locate the definition you want to modify in the table in detail-view and click on it.

The **Transaction Filter** panel appears in detail-view listing the current settings for the transaction filter definition you selected.

3 Click the **Modify** button.

The transaction filter parameters you can modify become editable in detail-view. For information on modifying these parameters, read the description of *Adding Transaction Filter Definitions*, elsewhere in this section.

4 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the changes.

Copying Transaction Filter Definitions

To use Event Replicator Administration to copy a transaction filter definition in the Replicator system file:

1 List the transaction filter definitions in Event Replicator Administration, as described in *Listing Transaction Filter Definitions*, elsewhere in this guide.

The transaction filter definitions are listed in detail-view.

2 Locate the definition you want to copy in the table in detail-view and click on it.

The **Transaction Filter** panel appears in detail-view listing the current settings for the transaction filter definition you selected.

3 Click the **Copy** button.

A copy of the transaction filter definition is created and its parameter values appear in detail-view.

4 Specify a new, unique name for the copy of the transaction filter definition in the Value column for the **Transaction Filter Name** parameter.

5 If you wish, modify any other parameters for the new copy in detail-view. For information on modifying the parameters, read the description of *Adding Transaction Filter Definitions*, elsewhere in this section.

6 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the copy.

Deleting Transaction Filter Definitions

To use Event Replicator Administration to delete a transaction filter definition in the Replicator system file:

1 List the transaction filter definitions in Event Replicator Administration, as described in *Listing Transaction Filter Definitions*, elsewhere in this guide.

The transaction filter definitions are listed in detail-view.

2 Locate the definition you want to delete in the table in detail-view and click on it.

The **Transaction Filter** panel appears in detail-view listing the current settings for the transaction filter definition you selected.

3 Click the **Delete** button.

Note: If you want to delete a filter condition from the transaction filter definition, read *Modifying Transaction Filter Definitions* elsewhere in this section.

A confirmation panel appears verifying that you want to delete the definition. If you click **Yes** (indicating that you do want to delete the definition), the definition is deleted. If you click **No** (indicating that you do not want to delete the definition), the definition is not deleted.

Rules for Writing Filter Conditions

There are various things you should consider when creating filter conditions. This section describes them.

- Using Wildcards

So My Record Matches the Filter Conditions -- Now What?

Filter conditions are based on the values of fields (or partial fields) in an SFILE record definition. If a field or partial field meets all of the filter conditions specified, the record is selected. Once selected, the record will be either included or excluded from replication processing, based on what the transaction filter definition specifies. So selection of a record does not necessarily mean that it will be replicated -- merely that it passed the filter conditions specified by the transaction filter definition. If the transaction filter definition indicates that selected records should be excluded from replication, the record will not be replicated.

Transaction filter definitions indicate whether selected records are replicated or not via the `FRECORDS` initialization parameter in the DDKARTE statements of the Event Replicator Server startup job or via the **Exclude or Include Records** field on the Transaction Filter screen of the Adabas Event Replicator Subsystem.

Note: Include and exclude processing function in the same way for partial fields as for complete fields in your transaction filters.

Failed or Ignored Filter Conditions

A filter condition will be ignored if it cannot be evaluated. This can occur if the image to be tested (set by the `FSIMAGE` or `FTIMAGE` parameters) is not present for replication. The effect of this on filter processing varies, based on whether the filter occurs as part of include or exclude processing and, if it is included in a group of conditions, how the other conditions in the group are matched, failed, or ignored. This is best explained in a series of examples.

Note: The following four examples using an add command (N1) are also true for an initial-state record since an initial-state record contains only an after image. No before image is present for an initial-state record.

1. Suppose an add command (N1) adds a record containing field AB to which the following filter is applied:

```
FRECORDS=INCLUDE
  FFIELD='AB',FSIMAGE=BI
    FCOND=EQ
    FLIST='1916'
```

   In this case, the filter cannot be evaluated because only the after image is present for an add and the filter is for the before image (FSIMAGE=BI). So this filter is ignored and no test is done on the field to see if the before image is equal to "1916". Consequently the add transaction is not included in replication.

2. Likewise, a similar exclude filter is also ignored:

```
FRECORDS=EXCLUDE
  FFIELD='AB',FSIMAGE=BI
    FCOND=EQ
    FLIST='1916'
```

In this case, the filter cannot be evaluated because only the after image is present for an add and the filter is for the before image (FSIMAGE=BI). So this filter is ignored and no test is done on the field to see if the before image is equal to "1916". However, because this is an exclude filter, the add transaction is not *excluded* from replication; in other words, it is included in replication, regardless of whether or not the before image of the AB field was equal to "1916".

3. Now consider the following transaction filter using **multiple filter conditions** and include processing:

```
FILTER NAME=MYINCLF
  FRECORDS=INCLUDE
    FFIELD='BA',FSIMAGE=BI,FCOND=EQ,FLIST='AAAA'
    FFIELD='BB',FSIMAGE=AI,FCOND=EQ,FLIST='VVVV'
    FFIELD='BC',FSIMAGE=AI,FCOND=EQ,FLIST='XXXX'
```

If an add command (N1) is issued for a record containing the BA, BB, and BC fields, no before image is present for these fields -- only the after image. Therefore, the filter condition for BA is ignored because the filter is for the before image; the BA filter condition is treated as if it is not even specified. The add transaction, then, is only *included* in replication if both filters for fields BB and BC are true.

4. Finally, consider the following transaction filter using **multiple filter conditions**, exclude processing, and OR processing:

```
FILTER NAME=MYEXCLF
  FRECORDS=EXCLUDE
    FFIELD='BA',FSIMAGE=BI,FCOND=EQ,FLIST='AAAA'
    FFIELD='BB',FSIMAGE=AI,FCOND=EQ,FLIST='VVVV'
  OR
    FFIELD='CA',FSIMAGE=AI,FCOND=EQ,FLIST='EEEE'
    FFIELD='CB',FSIMAGE=AI,FCOND=EQ,FLIST='CCCC'
  OR
    FFIELD='DA',FSIMAGE=BI,FCOND=EQ,FLIST='OOOO'
    FFIELD='DB',FSIMAGE=BI,FCOND=EQ,FLIST='CCCC'
```

If an add command (N1) is issued for a record containing these fields, no before image is present for these fields -- only the after image. Therefore, the filter conditions for BA, DA, and DB are ignored because the filters are for the before image; these filter conditions are treated as if they are not even specified.

The add transaction, then, is only *excluded* in replication if the filter for BB is satisfied OR if both the filters for field CA and field CB are satisfied. Otherwise, the add transaction is included in replication.

Target (FLIST Parameter) Value Syntax

Target (`FLIST` parameter) values are the values to be compared to the source field (`FFIELD` parameter) using the condition type specified (`FCOND` parameter). When multiple values are being compared for a field, they must be specified in a comma-separated list.

Each value can be expressed in one of two ways:

- You can specify values as free-format text. This text can be any set of alphanumeric set of characters. If blanks are required in the value, you should enclose the value in single quotes.

  When the data in the text is all numeric with an optional leading "+" or "-" sign, it is flagged as a numeric value and will be handled differently depending on the source field type in the Event Replicator Server definitions.
- You can specify values as a combination of `A()` and `X()` constructs that enable you to enter data for the same variable in alphabetic format, hexadecimal format, or both, as required. If the element value starts with the string "A(" or "X(" it is treated as an `A()` or `X()` value. If the value does not start with one of these strings, the value is treated as free-format text.

This section describes rules specific to these different methods of specifying target values.

Free-Format Value Rules

The following rules apply to free-format values.

- Free-format values can be any sequence of alphanumeric data apart from the comma character itself.
- If a blank is required for the free-format value, specify the value in single quotes.
- If an apostrophe is required as part of a free-format value, double the apostrophe (for example,‘six o’’clock’).
- If the value consists of all numeric characters with an optional leading "+" or "-" sign, the value will be treated as numeric.
- If the value begins with a single asterisk (*), it is interpreted as a wildcard suffix (for example, '*xyz').
- If the value ends with a single asterisk, it is interpreted as a wildcard prefix (for example, 'abc*').
- If two asterisks are found together (**) in any location in the free-format value, they are interpreted as a single asterisk in the resulting data.
- If a single asterisk is found in the middle of the data, it is rejected as invalid.

Note: The asterisk wildcard can only be used if the condition for the filter expression is EQ (equal) or NE (not equal). They cannot be used for any other types of filter expression conditions.

A() and X(0) Format Value Rules

The following rules apply to A() and X() value specifications.

- The `A()` construct is specified using the following syntax:

  ```
  A(data)
  ```

  In this syntax, the *data* specified can be any alphanumeric characters, except the parentheses characters.

- The `X()` construct is specified using the following syntax:

  ```
  X(data)
  ```

  In this syntax, the *data* specified must be an even number of characters in the range X'F0' to X'F9' (i.e. 0 to 9) and x'C1' to X'C6' (i.e. A to F). Each pair of characters will represent the hexadecimal value for one byte in the resultant value.

- If a value starts with an `A()` or `X()` construct, the entire value must be specified using these constructs. You cannot mix them with free-format values.
- `A()` and `X()` constructs can be specified multiple times in the same value specification. They must always have matching opening and closing parentheses, or the entire value specification is treated as invalid.
- When the `A()` construct is used, the asterisk (*) wildcard character is treated in the same manner as for free-format values.
- When the `X()` construct is used, the X'5C' character (which represents an asterisk) is treated like any other hexadecimal character and is not interpreted as a wildcard.

Examples

In the following example, an FLIST value of "ABCDE" is specified:

```
FLIST='ABCDE'
```

In the following example, a numeric FLIST value of "12345" is specified:

```
FLIST='12345'
```

In the following example, a numeric FLIST value of "-678" is specified:

```
FLIST='-678'
```

In the following example, an FLIST value of "AB123" is specified:

```
FLIST='AB123'
```

In the following example, an FLIST value of "XyZ" is specified:

```
FLIST='A(XyZ)'
```

In the following example, an FLIST value of "SSS" (the alphabetic equivalent of X'E2E2E2') is specified:

```
FLIST='X(E2E2E2)'
```

In the following example, an FLIST value of "abc<<<def" is specified:

```
FLIST='A(abc)X(4C4C4C)A(def)'
```

In the following example, an FLIST value of "AX(E2E2E2)" is specified:

```
FLIST='AX(E2E2E2)'
```

In the following example, an FLIST value of "1A(BCD)" is specified:

```
FLIST='1A(BCD)'
```

In the following example, an FLIST value of "1A(BCD)" is specified:

```
FLIST='1A(BCD)'
```

In the following example, an FLIST value of "*abc*" ("abc" is the alphabetic equivalent of X'C1C2C3') is specified. Note that this FLIST value is open-ended because wildcards are specified:

```
FLIST='A(*)X(C1C2C3)A(*)'
```

In the following example, an FLIST value of "*def**" is specified. Note that the first asterisk specifies a wildcard, but the last two asterisks specify asterisk characters (the alphabetic equivalent of X'5C5C'):

```
FLIST='A(*def)X(5C5C)'
```

The following examples are invalid because they specify a wildcard asterisk in the middle of the values:

```
FLIST='*ABC*DEF*'
FLIST='X(F5F6)*X(F7F8)'
FLIST='X(F2)A(*)X(F4)'
```

The following examples are invalid because they specify invalid hexadecimal data:

```
FLIST='X(ABACFGZZAE)'
FLIST='X(ABC)'
```

The following example is invalid because it mixes free-format and hexadecimal data:

```
FLIST='X(AB)AB'
```

The following example is invalid because it misuses commas:

```
FLIST='X(ABAC),,A(123)'
```

The following example is invalid because it misuses parentheses in the A() construct:

```
FLIST='A(12(34))'
```

When You Can Specify Multiple Targets

You can only specify multiple targets if the condition operator is EQ (equal) or NE (not equal). The LT (less than), LE (less than or equal), GT (greater than), and GE (greater than or equal) operators logically assume a comparison of the field value to a single target value, so multiple target values are not allowed for these condition operators.

Since wildcards are essentially a concise way of specifying multiple targets, you can also only use wildcards when the condition operator is EQ or NE.

If your filter checks to see if the field value is equal to a list of target values, the field value need only be equivalent to *one* of the target values for the filter condition to be true. On the other hand, if your filter checks to see if the field value is not equal to a list of target values, the field value must not be equal to *any* of the target values for the filter condition to be true.

Examples

In the following example, records for which the after image of the AA field is equal to "1", "2", "3", or "4" are selected.

```
FFIELD='AA',FSIMAGE=AI
FCOND=EQ
FLIST='1,2,3,4'
```

In the following example, records for which the after image of the AA field is greater than "5" are selected.

```
FFIELD='AA',FSIMAGE=AI
FCOND=GT
FLIST='5'
```

In the following example, records for which the first three bytes of the after image of the BB field contain the characters "abc" are selected.

```
FFIELD='BB',FSIMAGE=AI
FCOND=EQ
FLIST='abc*'
```

In the following example, records for which the last three bytes of the after image of the BB field contain the characters "xyz" are selected.

```
FFIELD='BB',FSIMAGE=AI
FCOND=EQ
FLIST='*xyz'
```

In the following example, records in which *no* bytes of the after image of the BB field contain the characters "klm" are selected.

```
FFIELD='BB',FSIMAGE=AI
FCOND=NE
FLIST='*klm*'
```

The following example is invalid because it specifies multiple FLIST target values when the condition code is not EQ or NE.

```
FFIELD='AA',FSIMAGE=AI
FCOND=LE
FLIST='1,2,3,4'
```

The following example is invalid because it specifies a wildcard in the FLIST target value when the condition code is not EQ or NE.

```
FFIELD=’AA’,FSIMAGE=AI
FCOND=GT
FLIST=’*xyz’
```

How Multiple Filter Conditions Are Interpreted

You can specify multiple filter conditions within a single transaction filter definition. Unless otherwise grouped, all of the specified filter conditions must be true for a record to be selected. In other words, the filter conditions are logically ANDed. In the following example, cond1, cond2, cond3, and cond4 must all be true for the record to be selected as they are logically ANDed:

```
FILTER NAME=MYINCLF
FRECORDS=INCLUDE
FFIELD='field',FCOND=cond1,FTARGET or FLIST values1
FFIELD='field',FCOND=cond2,FTARGET or FLIST values2
FFIELD='field',FCOND=cond3,FTARGET or FLIST values3
FFIELD='field',FCOND=cond4,FTARGET or FLIST values4
```

If, however, you want to insert some logical ORs in this example, you can. To do this you would use the OR keyword in the DDKARTE statements of the Event Replicator Server startup job or use the **Group** field on the Filter Condition screen in the Adabas Event Replicator Subsystem. As an example of using the OR keyword, consider the following modification to the example given earlier.

```
FILTER NAME=MYINCLF
FRECORDS=INCLUDE
FFIELD='field',FCOND=cond1,FTARGET or FLIST values1
FFIELD='field',FCOND=cond2,FTARGET or FLIST values2
FFIELD='field',FCOND=cond3,FTARGET or FLIST values3
OR
FFIELD='field',FCOND=cond4,FTARGET or FLIST values4
```

In this example, condition1, condition2, and condition3 must be true OR condition 4 must true for the record to be selected.

When using the **Group** field on the Filter Condition screen of the Adabas Event Replicator Subsystem to define your transaction filter definitions, simply use the same group number for those conditions you want ANDed. Conditions with different group numbers are logically ORed.

Specifying a Range of Values

You can specify a range of values in your filter condition by creating two conditions that are logically ANDed (read ***How Multiple Filter Conditions Are Interpreted***). Simply define one filter condition to test for values greater than (GT) or greater than or equal to (GE) the lowermost value. Then define the second filter condition to test for values less than (LT) or less than or equal to (LE) the uppermost value. As both conditions must be true since they are logically ANDed, your range specification is assured.

Field Type Considerations

Ideally, when a field is compared to another field, the field types will be the same. However, it is possible to compare fields of different formats. For example, you can compare a packed decimal format field with a binary format field. For a complete list of compatible Adabas field types, refer to your Adabas documentation.

This section covers the following topics related to how fields of different formats are compared

- Valid Comparison Table
- Comparison Processing by Field Type
- UES Considerations

Valid Comparison Table

An asterisk (*) in a cell in the following table indicates that a comparison of the field types is valid. A blank in a cell in the table indicates that a comparison is not supported.

Field Data Type	Alphanumeric	Unpacked	Packed	Binary	Floating Point	Wide-Character	Fixed Point
Alphanumeric	*			*		*	
Unpacked		*	*	*	*		*
Packed		*	*	*	*		*
Binary	*	*	*	*	*		*
Floating Point		*	*	*	*		*
Wide-Character	*					*	
Fixed Point		*	*	*	*		*

Comparison Processing by Field Type

When either the source or target field is of type floating point (but not both fields), the other field will be converted to floating point, and a floating point comparison will be made. SARC settings governing byte-swapping and floating point type (HFP, IEEEfloat, and VAXfloat) are honored.

Note: The conversion of very large numbers in a numeric format other than floating point to floating may result in a loss of precision because as the numbers get bigger, the range of numbers that may be represented in the floating point format is reduced. For example, the value 99,999,999,999,999,999 will be converted to the floating point value 99,999,999,999,999,984.

In all other cases the following conversions and comparisons will apply:

Source Field Data Type	Comparison Processing Notes
Unpacked	The source and target fields are converted to packed form for comparison.
Binary	Prior to comparison, the SARC byte order setting is honored for binary source and target fields. Packed and unpacked target fields are converted to binary and then compared. An alphanumeric target field is compared as is.
Packed	Prior to comparison, target fields of type fixed, unpacked, or binary are converted to packed.
Fixed	When the target field is binary, the source is converted to binary and then compared. When the target field is packed, the source field is converted to packed and then compared. When the target field is unpacked, both the source and target fields are converted to packed and then compared. When the target field is fixed, a direct comparison is made between the source and target fields (no conversion is necessary).

When a field is compared to a list of target values, the target values are converted (if they are not the same) to the data type of the source field, once the source field type is determined. This can cause problems in the accuracy of filter condition processing if a target value in the list cannot be converted or is otherwise incompatible with the required source field type. So target values and target fields must be specified carefully to avoid such problems.

Target list values entered as alphanumeric are converted to the data type of the source field, honoring the SARC, SACODE and SWCODE parameter settings.

Target list values for alphanumeric fields may be entered as alphanumeric, hexadecimal, or a mixture of both. If it is a mixture of both -- for example, FLIST=A(ABC)X(C4C5C6)A(GHI) -- it is treated as an alphanumeric field even though some of it is specified as hexadecimal.

Target list values for binary fields may be entered in hexadecimal. The hexadecimal values are assumed to be in a form that honors the SARC parameter settings.

Target list values for floating point fields may be entered in hexadecimal. The hexadecimal values are assumed to be in a form that honors the SARC parameter settings such as floating-point format and byte order.

Target list values may not be entered in hexadecimal for zoned decimal, packed decimal and fixed point fields.

UES Considerations

When a field is compared to a target value that is entered in hexadecimal, the target value is normally accepted without any conversion. It is assumed that you have taken into account the settings of the SARC, SACODE and SWCODE parameters when constructing the hexadecimal value. It is important to remember that a given hexadecimal value may have to reflect the setting of more than one of these three parameters.

- The SARC parameter defines special data architecture for fields in the record and value buffers (see the description of record buffers in your Adabas documentation). If the byte order bit of the SARC value is set, the hexadecimal value may have to be entered with low-order bytes first. If the field is a character field, the entered hexadecimal byte values must reflect the setting of the SARC encoding family bit.

  Note: If you want to transfer replicated and initial-state data to a relational database using the Event Replicator Target Adapter (the Destination Class, or DCLASS parameter, is set to "SAGTARG"), set the SARC parameter to "2" -- regardless of the location of your relational database.

- The SACODE parameter assigns special encoding for alphanumeric fields during the user session. Hexadecimal bytes values must reflect the SACODE setting.
- The SWCODE assigns special encoding for wide-character fields during the user session. Hexadecimal values must reflect the SWCODE setting.

This section covers the following topics related to UES processing:

- Internal Handling of UES Settings

Internal Handling of UES Settings

When FILTER FLIST value parameters are being processed, the UES settings and the FCOND values are taken into account in an attempt to minimize conversion overhead at runtime.

- If the FCOND setting is either EQ or NE then the FLIST value will be stored as entered. At runtime, this FLIST value can be compared directly with the field value with no conversion required regardless of the UES settings.
- If the FCOND setting is LT, GT, LE, or GE, the FLIST value will be converted, taking into account the UES settings, so that valid comparisons can be made for the specified FCOND value. At runtime, the field value will be similarly converted to facilitate valid comparisons.

It is important to note here that FLIST values that are entered as hexadecimal values for comparison with binary fields must be entered in a form that honors the SARC settings. In other words, they must be specified in the same form as the field is stored in the record buffer. Similarly wide-character field values entered as hexadecimal must be specified in the same form as the field is stored in the record buffer.

FLIST values that are entered as EBCDIC text, or as numbers, will be converted appropriately.

Examples Honoring UES Settings

Field Data Type	Examples

Floating Point	In the following example, an FLIST value of a short HFP floating point value of 1.0 is specified in hexadecimal:
	
	`FLIST='X(41100000)'`
	
	Regardless of the setting of the SARC byte order bit the hexadecimal value for a HFP floating point value will remain the same because HFP floating point does not honor byte swapping.
	
	In the following example, an FLIST value of a long HFP floating point value of 50,000.0 is specified in hexadecimal:
	
	`FLIST='X(44C3500000000000)'`
	
	In the following example, an FLIST value of a short VAX floating point value of 1.0 is specified in hexadecimal:

Field Data Type	Examples

	`FLIST='X(40800000)'`
	
	This value is valid when the `SARC` byte order bit is not set. If the SARC order bit is set, the following hexadecimal value must be entered for a value of 1.0:
	
	`FLIST='X(80400000)'`
	
	Note that the bytes are swapped in pairs.
	
	In the following example, an `FLIST` value of a long VAX floating point value of 50,000.0 is specified in hexadecimal:
	
	`FLIST='X(4843500000000000)'`
	
	This value is valid when the `SARC` byte order bit is not set. If the `SARC` order bit is set, the following hexadecimal value must be entered for a value of 50,000.0:
	
	`FLIST='X(4348005000000000)'`
	
	Note that the bytes are swapped in pairs.

Wide-character	In the following example, an `FLIST` value of the wide-character string '`ABB ABCDEF`' in WCODE=4095 is specified in hexadecimal:
	
	`FLIST='X(00410042004200400041004200430044004500460)'`
	
	In the following example, an `FLIST` value of the wide-character string '`ABB ABCDEF`' in WCODE=4095 with the `SARC` byte order bit turned on, is specified in hexadecimal:
	
	`FLIST='X(41004200420040004100420043004400450046 00)'`
	
	Note that the bytes are swapped in pairs.

Binary	In the following example, an `FLIST` value of the binary value of decimal 4 with the `SARC` byte order bit turned off, is specified in hexadecimal:
	
	`FLIST='X(000004)'`
	
	In the following example, an `FLIST` value of the binary value of decimal 4 with the `SARC` byte order bit turned on, is specified in hexadecimal:
	
	`FLIST='X(040000)'`
	
	Note that the byte order is reversed. In other words, the first byte becomes the last byte, the second byte becomes the second-to-last byte, and so on.

Varying Field Length Considerations

When the length of the source field and target field are different, the shorter value is converted to the size of the longer value. For alphanumeric data, the value is padded on the right with blanks. For numeric data, the value is padded on the left with hexadecimal zeros.

Using Wildcards

You can use an asterisk (*) as a wildcard for target values if the condition code being used is EQ (equal) or NE (not equal). You cannot use wildcard characters for any other filter conditions (GT, LT, LE, or GE).

Note: Wildcard values are not supported for wide character fields.

- If you want to test the field for any value beginning with a specific string of characters, simply append an asterisk to the end of the value. For example, to test for a field value starting with the characters "POW", specify "POW*" as the target value.
- If you want to test the field for the occurrence of as specific string within its value, precede and supercede the string with an asterisk. For example, to test for the occurrence of the string "WER", specify "*WER*" as the target value.
- If you need to test for the occurrence of an asterisk itself in a field value, specify two asterisks in a row for the target value ("**").

10 Maintaining Subscription Definitions

A subscription definition defines a set of rules to be applied to replicated data. Subscription definitions include at least one SFILE definition and at least one destination definition. They may also specify a resend buffer definition to expedite the retransmission of a transaction.

At least one subscription definition, with its associated destination and SFILE definitions, must be created because these definitions are used to determine how replicated data is processed by the Event Replicator Server. If a subscription definition is not specified, data replication will occur, but the data will never be processed by the Event Replicator Server and, therefore, will never be delivered to the target application.

Listing Subscription Definitions

To use Event Replicator Administration to list the subscription definitions stored in the Replicator system file:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Replication Definitions** in tree-view under the selected database.

3 Click on **Subscriptions** in the tree-view under **Replication Definitions**.

A table listing the subscription definitions in the Replicator system file appears in detail-view.

Adding Subscription Definitions

To use the Event Replicator Administration to add a subscription definition in the Replicator system file, complete the following steps:

Step 1. Access the Subscription Definition Area of the Event Replicator Administration

To access the subscription definition are of the Event Replicator Administration:

1 List the subscription definitions in Event Replicator Administration, as described in ***Listing Subscription Definitions***, elsewhere in this guide.

The subscription definitions are listed in detail-view.

2 Right-click on **Subscriptions** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3 Click on **Create New Subscription** in the drop-down menu.

A blank **New Subscription** panel appears in detail-view.

Note: The list of incomplete items on this screen identifies information (definitions) that must be supplied for the subscription definition before it can be read at Event Replicator Server startup. In the example below, the new subscription definition still needs the following information before it is a valid subscription: at least one destination definition, at least one SFILE definition (File-Related Parms), and format buffer specifications (either as individual format buffers or using global format buffer (GFB) definitions).

General Values	
Name	Value
Subscription Name	*
Description	*
User Data Alpha Key	0
Architecture Key	2 - High Order Byte First, EBCDIC Encoding Family, IB
Subscription Version	
User Data Wide Key	0
Resend Buffer Name	None
Subscription Active	Yes
Subscription Deactivate	Yes
Increment Initial State Count	No
Destination Name List	
File-Related Parameters	
Incomplete Item(s)	Destination, File-Related Parameters, Format Buffer

TLOG Values	
Name	Value
Input Filter	0 - No Logging
Filter Level	0 - No Logging
Output Filter	0 - No Logging
Filter Matched	0 - No Logging
Filter Not Matched	0 - No Logging
Filter Ignored	0 - No Logging

Step 2. Supply General Subscription Information

To supply general information for the subscription definition:

- Supply values for the following general subscription fields on the **New Subscription** panel:

Parameter Name	Specify	Default
Subscription Name (SUBSCRIPTION NAME)	A unique name for the subscription. Subscription names must be between one and eight characters long.	---
Description	A description of the subscription. This information is for your reference only; it is not used by Event Replicator Server.	----
User Data Alpha Key (SACODE)	An appropriate output-alpha code if the Event Replicator Server has been started with Universal Encoding Support (UES) enabled. If UES has not been enabled, leave this field blank or set to zero (0). A value of zero means that no UES translation will be done. XML data sent to Event Replicator Target Adapter is in EBCDIC character encoding. EBCDIC values for alphanumeric characters X'40' thru X'FF' are sent unchanged. EBCDIC values for alphanumeric fields under X'40' are translated to spaces. However, support for UTF-8 encoding is available for internationalization purposes. To send XML messages to Event Replicator Target Adapter using UTF-8 format, make sure that this parameter is set to "4091" in the Event Replicator subscription definition used for Event Replicator Target Adapter processing. **Note:** If you want to use UTF-8 character encoding (i.e. SACODE=4091), you must verify that your field lengths are increased as required to accommodate UTF-8 character encoding. The Adabas UES code pages 37, 424, 813, 912, 915, 920, 922, 923, 1006, 1112, 1140, 1256, and 4091 can be specified in the SACODE parameter for Event Replicator Target Adapter.	0
Architecture Key (SARC)	An appropriate architecture key if the Event Replicator Server has been started with Universal Encoding Support (UES) enabled. If UES has not been enabled, leave this field set to "2". The architecture key is an integer that is calculated as the sum of the following numbers (this is the same as is documented in the record buffer section of the OP command in the Adabas Command Reference Guide): ■ byte order: 0 (high-order byte first) or 1 (low-order byte first) ■ encoding family: 0 (ASCII) or 2 (EBCDIC) ■ floating point format: 0 (IBM370), 4 (VAX), or 8 (IEEE). **Note:** If you want to transfer replicated and initial-state data to a relational database using the Event Replicator Target Adapter (the Destination Class, or DCLASS parameter, is set to "SAGTARG"), set the SARC parameter to "2" -- regardless of the location of your relational database.	2 (High-order byte first, EBCDIC encoding family)

Parameter Name	Specify	Default
Subscription Version (SVERSION)	A subscription version. Up to two characters can be specified. This field has no meaning to the Event Replicator Server -- it is optional. But it may be useful for the target application when handling changes in the subscription definition. It is passed as part of the header information sent to the target.	---
User Data Wide Key (SWCODE)	An appropriate output-wide code if the Event Replicator Server has been started with Universal Encoding Support (UES) enabled. If UES has not been enabled, leave this field blank. Wide-character fields sent to Event Replicator Target Adapter are translated to hexadecimal. However, support for UTF-8 encoding is available for internationalization purposes. To send wide-character fields to Event Replicator Target Adapter using UTF-8 format, make sure that this parameter is set to "4091" in the Event Replicator subscription definition used for Event Replicator Target Adapter processing.	---
Resend Buffer Name (SRESENDBUFFER)	The name of the resend buffer to be associated with the subscription, if any. For more information about resend buffer definitions, read *Maintaining Resend Buffer Definitions*, elsewhere in this guide.	---
Subscription Active (SACTIVE)	Whether or not this subscription definition should be activated for use once it is loaded by the Event Replicator Server. Valid values are "Yes" (load and activate the definition) or "No" (load, but do not activate the definition).	No
Subscription Deactivate (SDEACTIVATE)	Whether or not this subscription definition should be deactivated if one of its files is deactivated. Valid values are "Yes" (deactivate the subscription) or "No" (do not deactivate the subscription). While this is an optional feature, we recommend that you use the default or specify "Yes" to ensure that the integrity of your replicated data is maintained when a file becomes deactivated. If the subscription is deactivated because one of its files is deactivated, any other files referenced by this subscription are also deactivated unless they are also referenced by another active subscription. If replication for this file in the subscription is unrelated to the replication for other files in the same subscription, the integrity of the replicated data may not be at risk. In this case, a value of "No" might be specified.	Yes
Increment Initial State Count (SINCREMENTIS)	Whether or not the transaction sequence number should be incremented if an initial-state request is occurring. Valid values are "Yes" (increment the transaction sequence number) or "No" (do not increment the transaction sequence number).	No

Step 3. (Optional) Modify the Transaction Logging Values for the Subscription, as Necessary

To modify the TLOG values for the subscription:

- Optionally, modify the following transaction logging (TLOG) fields on the **New Subscription** panel.

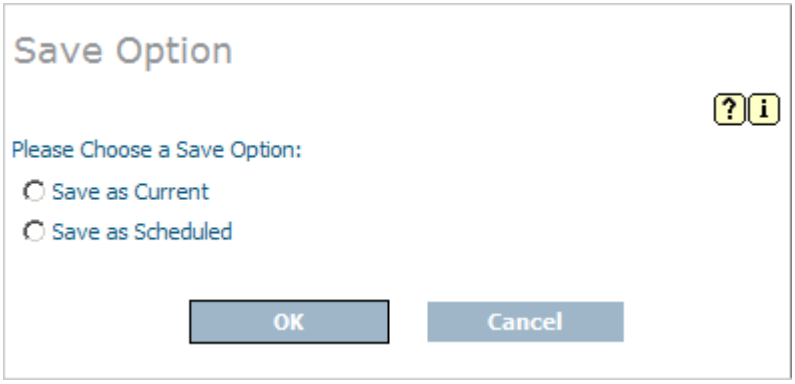
Parameter Name	Specify	Default
Input Filter (STLINPUT)	The transaction logging level when a transaction is selected for subscription processing. Valid values are "0 - no logging", "1- log event and input transaction data", "2- log event, input transaction, and file/record data", or "3- log event and all available input transaction data for the event".	0 - no logging
Filter Level (STLFILTER)	The transaction logging level when a record in a transaction is being excluded from replication due to extended subscription processing or the subscription user exit. Valid values are "0- no logging", "1- log event, filter reason, and transaction information", "2- log event, filter reason, transaction, and file/record information", or "3- log event, filter reason, transaction information, file/record information, and payload data of available images".	0 - no logging
Output Filter (STLOUTPUT)	The transaction logging level when a transaction is to be output on behalf of a subscription. Valid values are "0- no logging", "1- log event and output transaction data", "2- log event, output transaction, and file/record data", or "3- log event and all available output information for the event".	0 - no logging
Filter Matched (STLMATCH parameter)	The transaction logging level when a filter condition is true. Valid values are "0 - no logging", "1 - log event and filter information", or "2 - log event, filter information, and payload data or available field values".	0 - no logging
Filter Not Matched (STLNOMATCH parameter)	The transaction logging level when a filter condition is false. Valid values are "0 - no logging", "1 - log event and filter information", or "2 - log event, filter information, and payload data or available field values".	0 - no logging
Filter Ignored (STLIGNORE parameter)	The transaction logging level when a filter condition cannot be evaluated. Valid values are "0 - no logging", "1 - log event and filter information", or "2 - log event, filter information, and payload data or available field values".	0 - no logging

Step 4. Specify One or More Destinations for the Subscription

To specify one or more destinations for the subscription:

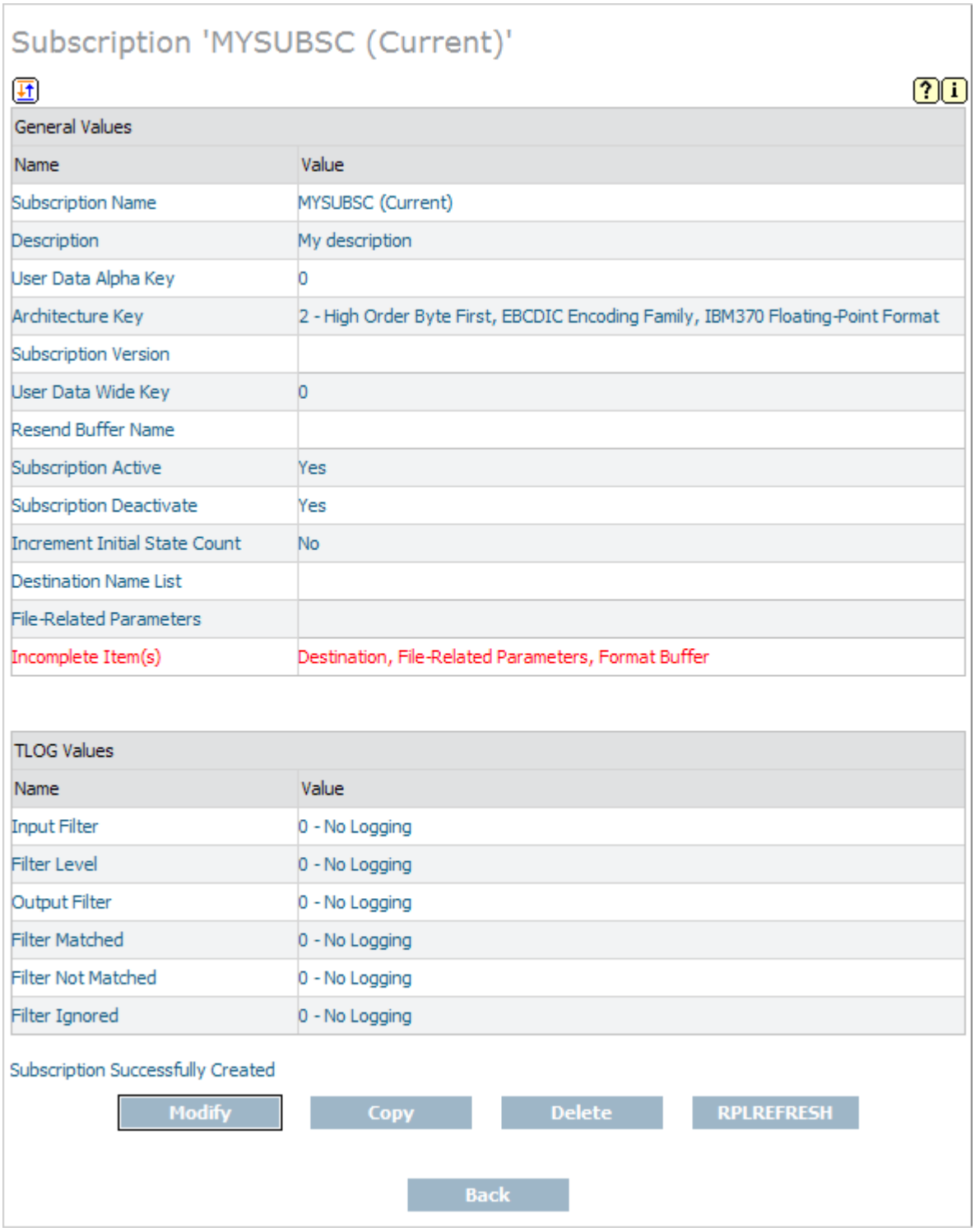
1 Click **OK** on the **New Subscription** panel. (Click **Cancel** if you want to cancel creating the subscription definition.)

A **Save Option** panel appears in detail-view.



2 Indicate whether you want to save the subscription definition as the current definition or as a scheduled definition on the Save Option panel. Then click **OK** to continue. (Click **Cancel** if you want to cancel creating the subscription definition.)

The **Subscription** panel appears again in detail-view.



Subscription 'MYSUBSC (Current)'

General Values	
Name	Value
Subscription Name	MYSUBSC (Current)
Description	My description
User Data Alpha Key	0
Architecture Key	2 - High Order Byte First, EBCDIC Encoding Family, IBM370 Floating-Point Format
Subscription Version	
User Data Wide Key	0
Resend Buffer Name	
Subscription Active	Yes
Subscription Deactivate	Yes
Increment Initial State Count	No
Destination Name List	
File-Related Parameters	
Incomplete Item(s)	Destination, File-Related Parameters, Format Buffer

TLOG Values	
Name	Value
Input Filter	0 - No Logging
Filter Level	0 - No Logging
Output Filter	0 - No Logging
Filter Matched	0 - No Logging
Filter Not Matched	0 - No Logging
Filter Ignored	0 - No Logging

Subscription Successfully Created

Modify Copy Delete RPLREFRESH

Back

3 Click the **Modify** button.

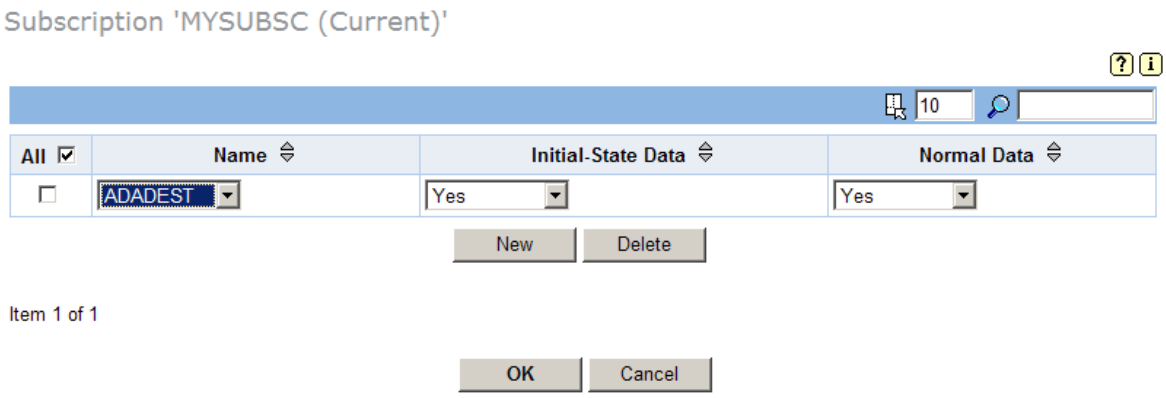
The parameters you can modify become editable again in detail-view.

General Values	
Name	Value
Subscription Name	MYSUBSC (Current)
Description	My description *
User Data Alpha Key	0
Architecture Key	2 - High Order Byte First, EBCDIC Encoding Family, IB
Subscription Version	
User Data Wide Key	0
Resend Buffer Name	None
Subscription Active	Yes
Subscription Deactivate	Yes
Increment Initial State Count	No
Destination Name List	✓
File-Related Parameters	✓
Incomplete Item(s)	Destination, File-Related Parameters, Format Buffer

TLOG Values	
Name	Value
Input Filter	0 - No Logging
Filter Level	0 - No Logging
Output Filter	0 - No Logging
Filter Matched	0 - No Logging
Filter Not Matched	0 - No Logging
Filter Ignored	0 - No Logging

4 Click on the checkmark in the **Value** column of the **Destination Name List** field.

The following destination list table appears in detail-view.



5 Specify at least one destination for the subscription in the destination list table. The first time you access this panel, no destinations are listed in the table.

- To add a new destination, click **New**. A new, blank, line appears in the destination list table. Select values in the new line as described later in this step.
- To modify a destination, edit the values for the destination in the table. Each destination value is described later in this step.
- To delete a destination, select the destination you want to delete by clicking the check box for the destination in the **All** column of the **Destination List** table. More than one destination can be selected at a time. To select all destinations at once, click the check box in the **All** column heading. Once all the destinations you want to delete are selected, click **Delete** to delete the selected destinations.

The fields in the **Destination List** table should be specified as follows:

Parameter Name	Use the selection box to select	Default
Name (SDESTINATION)	The name of a predefined destination definition. For more information about destination definitions, read *Maintaining Destination Definitions*, elsewhere in this guide. **Note:** An Adabas destination can be referenced by no more than one subscription.	---
Initial-State Data	Whether you want the destination to receive initial-state (I) replicated data. Valid values are "Yes" (receive the data) or "No" (do not receive the data).	Yes
Normal Data	Whether you want the destination to receive normal (N) data replicated data. Valid values are "Yes" (receive the data) or "No" (do not receive the data).	Yes

6 When all destinations have been specified correctly in the destination list table, click **OK** to save the subscription definition.

The destination list table panel closes and you are returned to the **Subscription** panel.

Step 5. Specify One or More SFILE Definitions for the Subscription

Specify an "S" in the **File-Related Parameters** field to create the SFILE definitions for the subscription. When you enter an "S", a List of Subscription SFILEs panel appears listing the SFILE definitions currently assigned to the subscription definition. At least one SFILE definition must be specified on the List of Subscription SFILEs screen:

- If the SFILE definition you want included in the subscription is not listed, click **New** to add it. If you want to alter an SFILE definition in this list, simply change the fields associated with that SFILE definition. For more information about maintaining SFILE definitions, read *Maintaining SFILE Definitions*, elsewhere in this guide.
- Once all of the SFILE definitions you want included in this subscription are listed on the List of Subscription SFILEs, press **OK** to return to rest of the subscription definition.

Step 6. Save the Subscription Definition

To save the subscription definition:

- Click **OK** to save the subscription definition in the Replicator system file.

Modifying Subscription Definitions

To use Event Replicator Administration to modify a subscription definition in the Replicator system file:

1. List the subscription definitions in Event Replicator Administration, as described in *Listing Subscription Definitions*, elsewhere in this guide.

   The subscription definitions are listed in detail-view.

2. Locate the definition you want to modify in the table in detail-view and click the check mark in either the **Current**, **Scheduled**, or **Old** columns, depending on which version of the definition you want to maintain.

   The detail-view lists the current settings for the subscription definition you selected.

3. Make sure that **Modify** is selected in the drop-down box at the bottom of detail-view. Then click **Execute**.

   The subscription parameters you can modify become editable in detail-view. For information on modifying these parameters, read the description of *Adding Subscription Definitions*, elsewhere in this section.

4 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the changes.

Copying Subscription Definitions

To use Event Replicator Administration to copy a subscription definition in the Replicator system file:

1 List the subscription definitions in Event Replicator Administration, as described in *Listing Subscription Definitions*, elsewhere in this guide.

The subscription definitions are listed in detail-view.

2 Locate the definition you want to copy in the table in detail-view and click the check mark in either the **Current**, **Scheduled**, or **Old** columns, depending on which version of the definition you want to copy.

The detail-view lists the settings for the version of the subscription definition you selected.

3 Click the **Copy** button.

A copy of the subscription definition is created and its parameter values appear in detail-view.

4 Specify a new, unique name for the copy of the subscription definition in the Value column for the **Subscription Name** parameter.

5 If you wish, modify any other parameters for the new copy in detail-view. For information on modifying the parameters, read the description of *Adding Subscription Definitions*, elsewhere in this section.

6 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the copy.

Activating and Deactivating Subscription Definitions

You can use Event Replicator Administration to activate and deactivate subscription definitions.

Caution: Be careful when you activate and deactivate replication definitions and databases, especially if replication is ongoing at the time. Whenever you activate or deactivate definitions or databases, you run the risk of altering what data is replicated and how that replication occurs. If the Event Replicator Server receives data from an Adabas database for which it has no active definitions, replication simply does not occur.

This section covers the following topics:

Activating Subscription Definitions

To use Event Replicator Administration to activate a subscription definition:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Subscriptions** in the tree-view under **Replication**.

A table listing the subscription definitions in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the definition you want to activate.

Details about the subscription appear in detail-view.

6 Click the **Activate** button.

Note: This button will not be available if the definition is already activated.

The subscription definition is activated.

Deactivating Subscription Definitions

To use Event Replicator Administration to deactivate a subscription definition:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the selected database.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

4 Click on **Subscriptions** in the tree-view under **Replication**.

A table listing the subscription definitions in the Replicator system file appears in detail-view.

5 In detail-view, click on the name of the definition you want to deactivate.

Details about the subscription appear in detail-view.

6 Click the **Deactivate** button.

Note: This button will not be available if the definition is already deactivated.

The subscription definition is deactivated.

Deleting Subscription Definitions

To use Event Replicator Administration to delete a subscription definition in the Replicator system file:

1 List the subscription definitions in Event Replicator Administration, as described in *Listing Subscription Definitions*, elsewhere in this guide.

The subscription definitions are listed in detail-view.

2 Locate the definition you want to delete in the table in detail-view and click the check mark in either the **Current**, **Scheduled**, or **Old** columns, depending on which version of the definition you want to maintain.

The detail-view lists the settings for the version of the subscription definition you selected.

3 Click the **Delete** button.

A confirmation panel appears verifying that you want to delete the definition. If you click **Yes** (indicating that you do want to delete the definition), the definition is deleted. If you click **No** (indicating that you do not want to delete the definition), the definition is not deleted.

11 Maintaining SFILE Definitions

An SFILE definition defines the Adabas file containing the input data to be processed by the subscription. SFILE definitions are included within subscription definitions. They include format buffer specifications for the file data (they can reference GFB definitions instead). SFILE definitions may also specify a transaction filter definition that should be applied to the data in the SFILE file and a subscription user exit that should be used in processing the file data.

At least one SFILE definition is required in a subscription definition for every Event Replicator for Adabas run.

Listing SFILE Definitions

To use Event Replicator Administration to list the SFILE definitions stored in the Replicator system file:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Replication Definitions** in tree-view under the selected database.

3 Click on **Subscriptions** in the tree-view under **Replication Definitions**.

A table listing the subscription definitions in the Replicator system file appears in detail-view.

4 Click on the subscription definition in detail-view that contains the SFILE definitions you want to list or maintain.

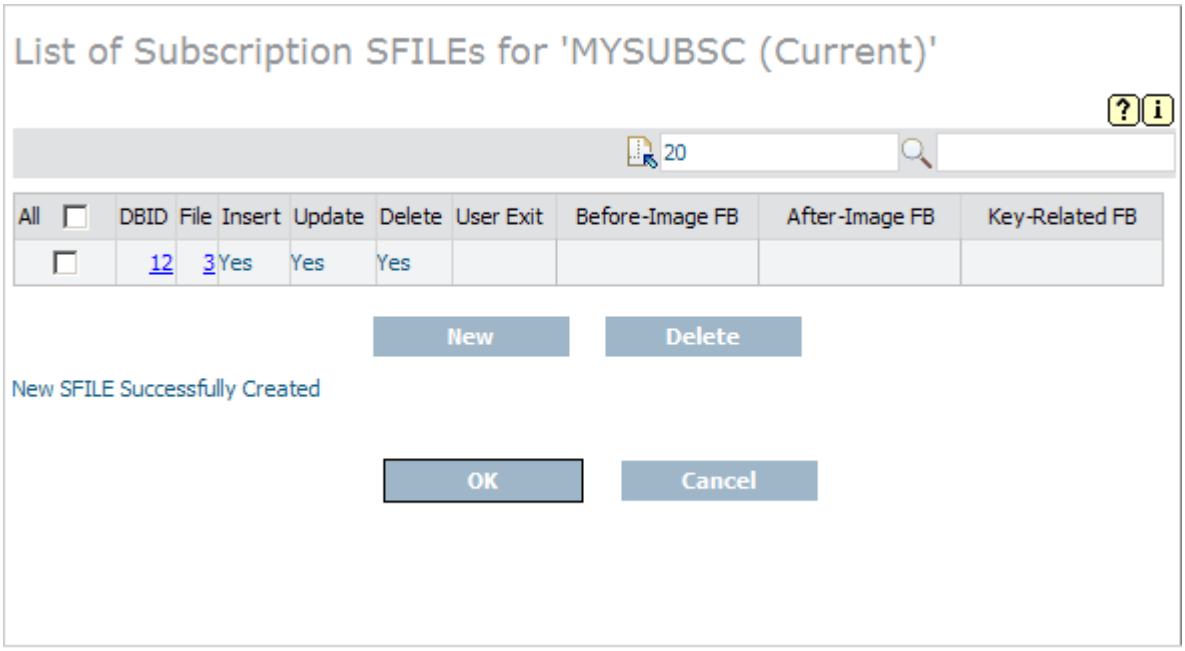
The subscription definition appears in detail view.

5 Click the **Modify** button in detail-view.

The subscription definition becomes modifiable.

6 Click on the check mark in the **File-Related Parameters** field in detail-view.

The **List of Subscription SFILEs** table for the subscription appear in detail-view.



Adding SFILE Definitions to a Subscription

To add SFILE definitions to a subscription, complete the following steps:

- Step 1. Access the SFILE Definition Area for the Subscription
- Step 2. Create the New SFILE Definition and Supply General SFILE Definition Information
- Step 3. Specify Format Buffers for the SFILE Definition
- Step 4. Save the SFILE Definitions
- Step 5. Save the Subscription Definition

Step 1. Access the SFILE Definition Area for the Subscription

To access the subscription definition are of the Event Replicator Administration:

- List the SFILE definitions in Event Replicator Administration, as described in *Listing SFILE Definitions*, elsewhere in this guide.

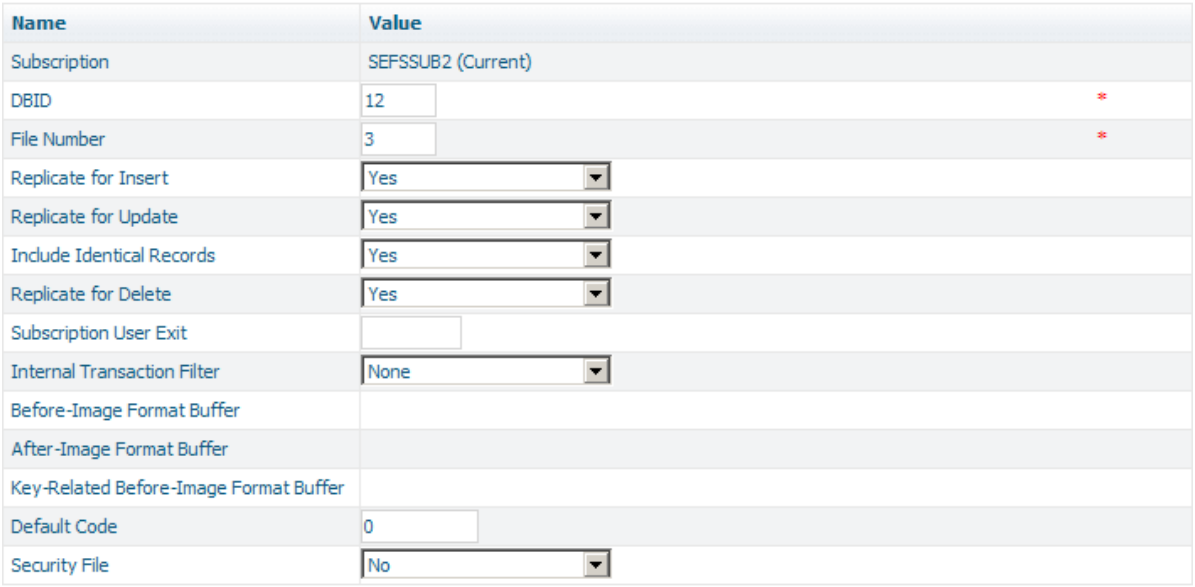
  The SFILE definitions for the subscription (if any) are listed in a **List of Subscription SFILEs** table in detail-view.

Step 2. Create the New SFILE Definition and Supply General SFILE Definition Information

To create the SFILE definition and supply general information for the SFILE definition:

1 Click the **New** button in detail-view.

A blank **SFILE** screen appears in detail-view.



2 Supply values for the following general SFILE definition fields on the **SFILE** panel:

Parameter Name	Specify	Default
Subscription	The name of the subscription definition you are editing. You cannot change this field here.	
DBID (SFDBID)	The database ID associated with the input file specified by this SFILE definition. The database ID is numeric and can range from one through 65535. This is a required field.	---
File Number (SFILE)	The number of the input file to be processed by this SFILE definition. This is a required field.	---
Replicate for Insert (SFREPLICATEINSERT)	Whether or not you want records from this Adabas file replicated when they are inserted. Specify "Yes" if you want them replicated or "No" if you do not want them replicated. This is an optional field.	Yes
Replicate for Update (SFREPLICATEUPDATE)	Whether or not you want records from this Adabas file replicated when they are updated. Specify "Yes" if you want them replicated or "No" if you do not want them replicated. This is an optional field.	Yes

Parameter Name	Specify	Default
Include Identical Records (SFREPLICATENOTCHANGED)	Whether an update request should be replicated if the after image (AI) is the same as the before image (BI). Valid values are "Yes" or "No". If "Yes" is specified and the before and after images are the same, the record is replicated subject to other field filtering that may be specified for this file. If "No" is specified, and the before and after images are the same, no further processing occurs for the record. Note that the format buffers for the before and after images must be identical. This is an optional field.	Yes
Replicate for Delete (SFREPLICATEDELETE)	Whether or not you want records from this Adabas file replicated when they are deleted. Valid values are "Yes", "No", or "Update". This is an optional field; the default is "Yes". If this parameter is set to "No" and an input record for the DBID/file is for a delete, the input record will NOT be processed for this subscription. If this parameter is set to "Yes" and an input record for the DBID/file is for a delete, the input record will be processed for this subscription. If this parameter is set to "Update" and an input record for the DBID/file is for a delete, the before and after images of the input record are passed to your subscription user exit. Therefore, if this parameter is set to "Update", a subscription user exit name must be specified in the Subscription User Exit parameter. In addition, the subscription before and after image format buffers must be identical and no primary key should be defined to the file (to ensure that the before image is a copy of data storage). The purpose of the "Update" value of this parameter is to allow your subscription user exit to process replicated physical delete transactions on your target database as you choose. Your subscription user exit can decide if the physical delete transaction should be: physically deleted from your target database, converted to an update, or ignored and not sent at all. For more information about using the subscription user exit with the "Update" setting of this parameter, read *Controlling Delete Transaction Processing (SFREPLICATEDELETE=UPDATE Processing)*, in the *Event Replicator for Adabas Administration and Operations Guide*.	Yes
Subscription User Exit (SFSEXIT)	The name of the subscription user exit you want used (if any) when records from the file specified by this SFILE definition are replicated. This is an optional field. For more information about subscription user exits, read *Using the Event Replicator Subscription User Exit*, in *Event Replicator for Adabas Administration and Operations Guide* provided with your Event Replicator Administration documentation.	---

Parameter Name	Specify	Default
Internal Transaction Filter (SFFILTER)	The name of the internal transaction filter definition you want used (if any) when records from the file specified by this SFILE definition are replicated. This is an optional field.	---
Before-Image Format Buffer	See the later steps in these instructions.	---
After-Image Format Buffer	See the later steps in these instructions.	---
Key-Related Before-Image Format Buffer	See the later steps in these instructions.	---
Default Code (SFDEFAULTACODE parameter)	A default, at the subscription file level, for the file's alpha character encoding. The encoding must belong to the EBCDIC encoding family; that is, the space character must be X'40'.'. This can only be set when the Event Replicator Server has been started with Universal Encoding Support (UES) enabled. This parameter is meant to be used when all of the following conditions are met: ■ The Event Replicator Server is UES-enabled, ■ The source database is not UES-enabled ■ The subscription definition requests UES translation (i.e. parameter SACODE is specified) ■ The data in the source database file is stored in a code page other than the default alpha encoding. At the time a before or after image is decompressed and translated in the Event Replicator Server, the file encoding is taken as follows: ■ from the source FCB; ■ if the encoding is not set above, from the GCB of the source database; ■ if the encoding is not set above, from the value set for this parameter (SFDEFAULTACODE); ■ if the encoding is not set above, from the GCB of the Event Replicator Server. Note that if the source database is UES-enabled, the file encoding will be taken from either a or b above.	none
Security File(SFSECURITYFILE parameter)	Whether or not you want security definitions in the Adabas security file replicated. Valid values are "Yes" or "No", with a default of "No". The setting of this parameter allows you to define a subscription file (SFILE) definition specifically for security definitions in the Adabas security file on the database. **Note:**	No

Parameter Name	Specify	Default
	1. This parameter cannot be set to "Yes" unless you have Adabas 8.2 or later installed. 2. Adabas Security Facilities, including the Adabas security utility (ADASCR) can be obtained only by special request. If you are interested in Adabas Security Facilities, please contact your Software AG sales representative. When this parameter is set to "No", replication of the security definitions does not occur and replication processing proceeds normally. When this parameter is set to "Yes", replication of the security definitions in the Adabas security file will occur. However, the following parameter settings are also required: ■ A format buffer may *not* be specified. This means that no values may be specified for the **Before-Image Format Buffer**, **After-Image Format Buffer**, or **Key-Related Before-Image Format Buffer** parameters. ■ No value may be specified for the **Default Code** parameter. ■ A transaction filter definition may *not* be specified. This means that no value may be specified for the **Transaction Filter** parameter. ■ The default value of `Yes` must be specified for the **Replicate for Insert**, **Replicate for Update**, **Include Identical Records**, and **Replicate for Delete** parameters. ■ A subscription exit may *not* be specified. This means that no value may be specified for the **Subscription User Exit** parameter. ■ When you set this parameter to "Y", you indicate that the file specified in the **File Number** parameter is the file number of the security file for the source database identified in the **DBID** parameter. Therefore, if this parameter is set to "Y" for other subscription SFILE definitions using the same source database (with the same SFILE DBID setting), the same value must be set for each of the **File Number** parameters in the SFILE definition. In other words, it is invalid for a source database (**DBID** setting) to have different file numbers specified for the security file in different subscriptions. For example, source database 10 cannot have the security file specified as both file 15 in one SFILE definition and 20 in another SFILE definition. ■ If this parameter is set to "Y" for a source database (DBID parameter), any Adabas destination definitions with same database specified as the destination input database, must also specify identical file numbers for both the input file and target file parameters.	

Parameter Name	Specify	Default
	For complete information about replicating security definitions, read *Replicating Security Definitions*, in the *Event Replicator for Adabas Administration and Operations Guide*.	

3 Click **OK** to save the incomplete SFILE definition.

The SFILE definition is saved for the subscription and appears in a **List of Subscription SFILEs** table in detail-view.

Step 3. Specify Format Buffers for the SFILE Definition

In each SFILE definition you create, you must specify a format buffer or a global format buffer (GFB) to be used when decompressing the data storage after image (an after-image format buffer). Format buffers or GFBs do not need to be specified for the before-image or key-related before-image format buffers.

To specify one or more SFILE definitions for the subscription:

1 Modify the settings of the following format buffer parameters, as appropriate for the SFILE definition. For information on defining global format buffers for use in SFILE definitions, read ***Maintaining GFB Definitions***, elsewhere in this guide.

Parameter Name	Specify	Default
Before Image Format Buffer	Using the drop-down list for this field, select a global format buffer (GFB) definition or select "Format Buffer" to type in a format buffer to be used when decompressing the data storage before image. Format buffer specifications must conform to the format buffer rules (for read commands) described in the Adabas command reference documentation (found in *Adabas Command Reference*). PE and MU fields cannot use the range notation 1-N in format buffers for a subscription that is sent to a destination that has specified the replication intialization parameter DCLASS=SAGTARG. The SAGTARG application invoked requires that the range of occurrences specified are contained in the record buffer even if they are empty occurrences. 1-N results in a range of 1-191, but unless there are 191 occurrences containing data, space in the record buffer is not allocated for any empty occurrences, resulting in incorrect field positioning when processing the record. If you are using the Event Replicator Target Adapter (if **"SAGTARG"** is specified as the destination class of some destination used by this subscription), any GFB names you select for the `After Image Format Buffer` parameter and the `Before Image Format Buffer` parameter	The after image format buffer will be used, if no before image format buffer is specified.

Parameter Name	Specify	Default
	must be the same. In addition, any GFB name you select for the `Key-Related Before Image Format Buffer` parameter must have been built from a Predict user view with the same name as the user view used to build the before and after images, but with the suffix "-KEY" on the end. (Likewise, if you use the Data Mapping Tool, the SGFORMATKEY GFB should be built from a DDM with the same name as the DDM used to build the before and after image GFBs.) Event Replicator processing strips off the "-KEY" to ensure that any delete is associated with the before and after image file name that was used to build the table(s) in the RDBMS.	
After Image Format Buffer	Using the drop-down list for this field, select a global format buffer definition or select "Format Buffer" to type in a format buffer to be used when decompressing the data storage after image. Format buffer specifications must conform to the format buffer rules (for read commands) described in the Adabas command reference documentation (found in *Adabas Command Reference*). If the destination target for replicated data is an Adabas mainframe database running Adabas 8.1.3 or later, you can specify the symbolic notation "C." alone as the After Image Format Buffer value or you can select the name of a GFB definition containing the symbolic notation "C." alone. This notation will cause the Event Replicator Server to replicate the entire source record to the target mainframe Adabas database. **Caution:** This "C." option may be used *only* if the destination target file has been defined with the same fields in the same order as the fields in the source file; if there are differences in the definitions of the files, the replication of the data is likely to incur errors. The only exception to this rule is that the definitions of the descriptors and superdescriptors in the target and source files may be different. PE and MU fields cannot use the range notation 1-N in format buffers for a subscription that is sent to a destination that has specified the replication intialization parameter DCLASS=SAGTARG. The SAGTARG application invoked requires that the range of occurrences specified are contained in the record buffer even if they are empty occurrences. 1-N results in a range of 1-191, but unless there are 191 occurrences containing data, space in the record buffer is not allocated for any empty occurrences, resulting in incorrect field positioning when processing the record. If you are using the Event Replicator Target Adapter (if **"SAGTARG"** is specified as the destination class of some destination used by this subscription), any GFB names you select for the `After Image Format Buffer` parameter and the `Before Image Format Buffer` parameter must be the same. In addition, any GFB name you select for the `Key-Related Before Image Format Buffer` parameter must have	No default. This is a required field. Either a format buffer or a GFB must be supplied.

Parameter Name	Specify	Default
	been built from a Predict user view with the same name as the user view used to build the before and after images, but with the suffix "-KEY" on the end. (Likewise, if you use the Data Mapping Tool, the SGFORMATKEY GFB should be built from a DDM with the same name as the DDM used to build the before and after image GFBs.) Event Replicator processing strips off the "-KEY" to ensure that any delete is associated with the before and after image file name that was used to build the table(s) in the RDBMS.	
Key-Related Before Image Format Buffer	Using the drop-down list for this field, select a global format buffer definition or select "Format Buffer" to type in a format buffer to be used when decompressing the key-related before image. Format buffer specifications must conform to the format buffer rules (for read commands) described in the Adabas command reference documentation (found in *Adabas Command Reference*). If you are using the Event Replicator Target Adapter (if **"SAGTARG"** is specified as the destination class of some destination used by this subscription), any GFB names you select for the `After Image Format Buffer` parameter and the `Before Image Format Buffer` parameter must be the same. In addition, any GFB name you select for the `Key-Related Before Image Format Buffer` parameter must have been built from a Predict user view with the same name as the user view used to build the before and after images, but with the suffix "-KEY" on the end. (Likewise, if you use the Data Mapping Tool, the SGFORMATKEY GFB should be built from a DDM with the same name as the DDM used to build the before and after image GFBs.) Event Replicator processing strips off the "-KEY" to ensure that any delete is associated with the before and after image file name that was used to build the table(s) in the RDBMS.	The field name followed by a period will be used as a default for this format buffer.

2 When the format buffers have been specified, click **OK**.

The **List of Subscription SFILEs** table appears.

Step 4. Save the SFILE Definitions

To save the SFILE definitions in the SFILE definition list for a subscription:

- Click OK to save the SFILE definitions in the Replicator system file.

Step 5. Save the Subscription Definition

To save the subscription definition:

- Click **OK** to save the subscription definition in the Replicator system file.

Modifying SFILE Definitions in a Subscription

To modify the SFILE definitions in a subscription:

1. List the SFILE definitions in Event Replicator Administration, as described in *Listing SFILE Definitions*, elsewhere in this guide.

   The SFILE definitions for the subscription (if any) are listed in a **List of Subscription SFILEs** table in detail-view.
2. Locate the SFILE definition you want to modify in the table in detail-view and click on either the database or file number of the definition.

   The detail-view lists the current settings for the SFILE definition you selected.
3. Modify the settings for the SFILE definition as described in *Adding SFILE Definitions*, elsewhere in this section.
4. When all modifications have been made, click **OK** to save the changes to the SFILE definition or click **Cancel** to cancel the changes.

   The **List of Subscription SFILEs** table appears.
5. Verify that at least one SFILE for the subscription is listed in the **List of Subscription SFILEs** table. Then click **OK** to save the list of SFILE definitions for the subscription.

   The general subscription information is listed in detail-view.
6. Click **OK** to save the subscription.

Deleting SFILE Definitions in a Subscription

To delete SFILE definitions included in a subscription definition:

1. List the subscription definitions in Event Replicator Administration, as described in *Listing SFILE Definitions*, elsewhere in this guide.

The SFILE definitions for the subscription (if any) are listed in a **List of Subscription SFILEs** table in detail-view.

2 Locate the SFILE definition you want to delete in the table in detail-view and click on the check box for the definition in the **All** column of the **List of Subscription SFILEs** table. To select all definitions at once, click on the check box in the **All** column heading.

3 Once all the definitions you want to delete are selected, click **Delete** to delete the selected SFILE definitions.

4 Verify that at least one SFILE for the subscription is listed in the **List of Subscription SFILEs** table. Then click **OK** to save the list of SFILE definitions for the subscription.

The general subscription information is listed in detail-view.

5 Click **OK** to save the subscription.

12 Maintaining GFB Definitions

A GFB definition defines a global format buffer (GFB) stored separately for reference in SFILE definitions. GFB definitions can be used to decompress replicated data from a specific database file for one or more subscriptions.

While a format buffer specification is required in a subscription's SFILE definition, a stored GFB definition does not need to be used. The SFILE definition could simply include the format buffer specifications it needs.

Using Event Replicator Administration you can manually add, modify, and delete GFB definitions. If you have an appropriate version of Predict installed (see *Predict Requirements* in the *Event Replicator for Adabas Installation Guide*), you can also generate a GFB. If the correct version of Predict is not installed, you will not be able to use this feature.

Notes:

1. If you will be using the Event Replicator Target Adapter to replicate data to an RDBMS, you must generate your GFB definitions, as described in ***Generating a GFB***, elsewhere in this chapter.
2. You can also use the Data Mapping Tool, provided with Event Replicator Administration, to generate a GFB and associated field table. For complete information on the Data Mapping Tool, read *Using the Data Mapping Tool*, in the *Data Mapping Tool User's Guide*.
3. If you want to use UTF-8 character encoding, you must verify that your GFB field lengths are increased as required to accommodate the character set referenced by the code page you select and the data requested in the GFB. You can increase these field lengths manually by editing the GFB itself or by editing the Predict file or data definition module (DDM) used when the GFB is generated.

Listing GFB Definitions

To use Event Replicator Administration to list the general format buffer (GFB) definitions stored in the Replicator system file:

1 Select an Event Replicator Server in tree-view as described in ***Selecting Event Replicator Databases***, elsewhere in this guide.

2 Click and expand **Replication Definitions** in tree-view under the selected database.

3 Click on **Global Format Buffers** in the tree-view under **Replication Definitions**.

A table listing the GFB definitions in the Replicator system file appears in detail-view.

Adding GFB Definitions

You can add global format definitions by typing in the format buffer definition manually or using Predict to generate one for you. This section describes how to add one manually.

To use Event Replicator Administration to add a global format definition manually to the Replicator system file:

1 List the GFB definitions in Event Replicator Administration, as described in *Listing GFB Definitions*, elsewhere in this guide.

The GFB definitions are listed in detail-view.

2 Right-click on **Global Format Buffers** in the tree-view under **Replication Definitions**.

A drop-down menu appears.

3 Click on **Create New Format Buffer** in the drop-down menu.

A blank **Create New Global Format Buffer** panel appears in detail-view.

4 Tab to Value of the **Global Format Buffer Name** field and specify a unique name for the global format buffer definition. The name must be between one and seven alphanumeric characters long.

Note: If, at this point, you want to generate the GFB and field table, rather than enter it manually, click on **Predict Parameters**. The Predict Parameters table appears in detail-view. For more information about generating a GFB, skip the rest of the documentation in this section and read *Generating a GFB* elsewhere in this section.

5 If you elect not to generate the GFB, use the box in the **Create New Global Format Buffer** panel to manually specify the format buffer for this definition. The format buffer must conform to the format buffer requirements (for read commands) documented in the Adabas command reference documentation (found in *Adabas Command Reference*).

Use the scroll bar to scroll backwards and forwards through the format buffer definition.

PE and MU fields cannot use the range notation 1-N in format buffers for a subscription that is sent to a destination that has specified the replication intialization parameter DCLASS=SAGTARG. The SAGTARG application invoked requires that the range of occurrences specified are contained in the record buffer even if they are empty occurrences. 1-N results in a range of 1-191, but unless there are 191 occurrences containing data, space in the record buffer is not allocated for any empty occurrences, resulting in incorrect field positioning when processing the record.

If the destination target for replicated data is an Adabas mainframe database running Adabas 8.1.3 or later, you can specify the symbolic notation "C." alone as the global format buffer. This notation will cause the Event Replicator Server to replicate the entire source record to the target mainframe Adabas database.

Caution: This "C." option may be used *only* if the destination target file has been defined with the same fields in the same order as the fields in the source file; if there are differences in the definitions of the files, the replication of the data is likely to incur errors. The only exception to this rule is that the definitions of the descriptors and superdescriptors in the target and source files may be different.

6 When all specifications have been made to your satisfaction, click **OK** to save the GFB definition.

Generating a GFB

If you have Predict 4.4.1 with Service Pack 3 (or higher) installed, you can now generate format buffers and corresponding field tables from Predict file definitions using the Event Replicator Administration. If this version of Predict is not installed, you will not be able to use this feature.

Notes:

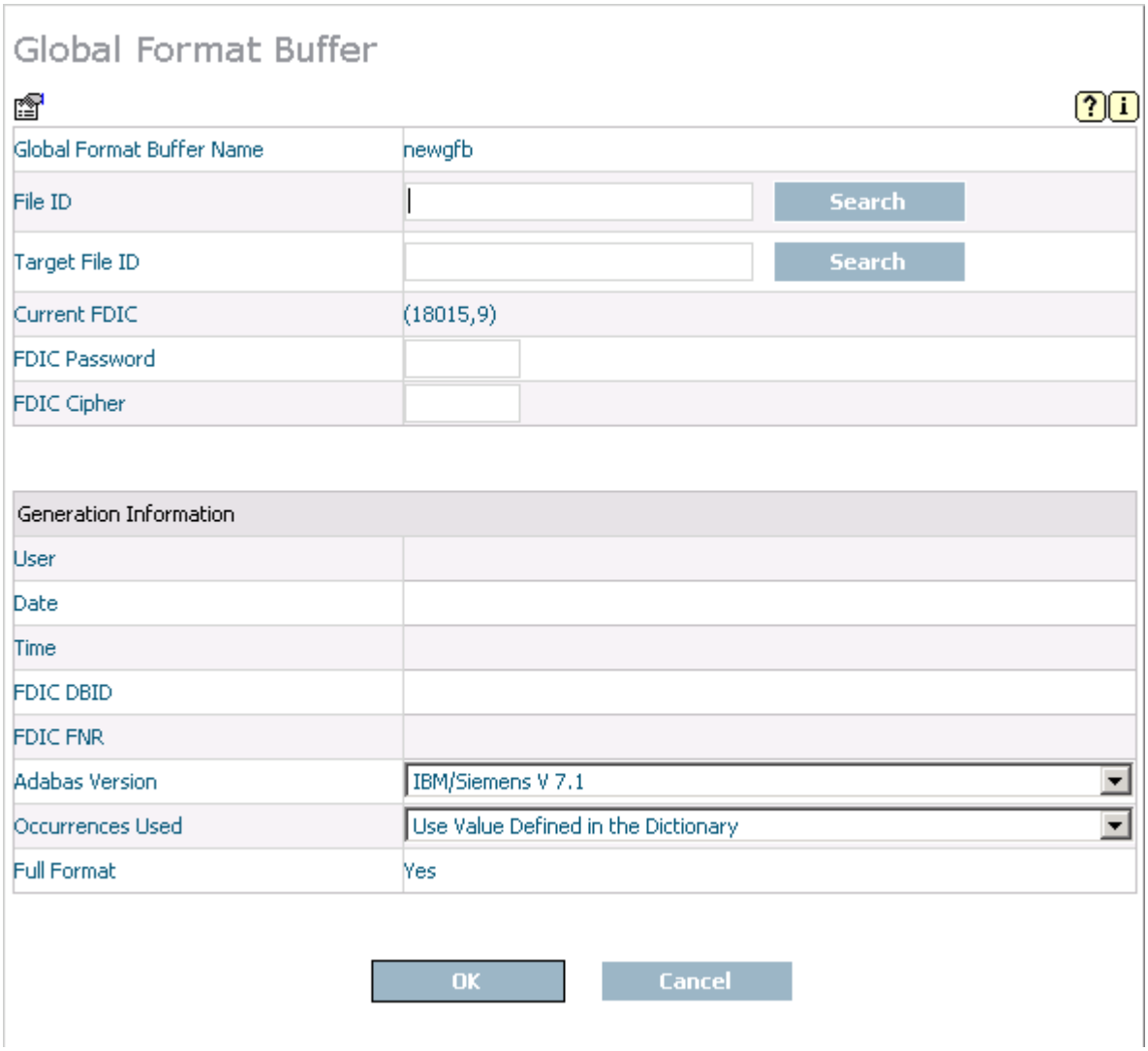
1. If you will be using the Event Replicator Target Adapter to replicate data to an RDBMS, you must generate your GFB definitions, as described in this section. Alternatively, you can import a GFB that was generated using the Data Mapping Tool. For information on importing a Data Mapping Tool-generated GFB, read ***Importing a Data Mapping Tool GFB and Field Table***, elsewhere in this section. For information on using the Data Mapping Tool, read *Using the Data Mapping Tool*, in *Data Mapping Tool User's Guide*.
2. MU and PE fields are supported in the generated format buffers and their corresponding field tables, although MU fields within PE fields are not. This support requires that counters for MU and PE fields be available in the Predict data dictionary definition. Counters for MU and PE fields are available if the Predict data dictionary definition either specifies fields with field types of MC (MU fields with an automatic counter) and PC (PE fields with an automatic counter) *OR* includes explicit CM field (MU counter field) or CP field (PE counter field) definitions. For more information about CM, CP, MC, MU, PC, and PE field types, refer to your Predict documentation.
3. PE and MU fields cannot use the range notation 1-N in format buffers for a subscription that is sent to a destination that has specified the replication intialization parameter DCLASS=SAGTARG. The SAGTARG application invoked requires that the range of occurrences specified are contained in the record buffer even if they are empty occurrences. 1-N results in a range of 1-191, but unless there are 191 occurrences containing data, space in the record buffer is not allocated for any empty occurrences, resulting in incorrect field positioning when processing the record.

4. LA fields are not supported in generated format buffers.

To use Event Replicator Administration to generate global format buffer definition using Predict:

1 Click on **Predict Parameters** while editing a GFB definition.

The following screen appears in detail-view.



2 Update the following fields in the screen as described in the following table.

Parameter Name	Specify	Default
Global Format Buffer Name	The unique name of the GFB definition you are editing. You cannot change this name here.	---
File ID	The name of a Predict file with a file type or Adabas (A) or Adabas user view (U). Use the **Search** button to search and select a file from a list.	---
Target file ID	The name of a Predict file with a file type of sequential (S). This file may be used to insert space notation (nX) into the format buffer. For fields in the target file with matching definitions in the File ID file (in other words, if a field exists with the same field long name), a short name clause is generated. For fields that do not have a matching definition in the File ID file, an appropriate space notation (nX clause) is generated. The spaces defined by nX clauses can be filled using a user exit. Use the **Search** button to search and select a file from a list.	---
Current FDIC	The database and file number of the current Predict file.	---
FDIC Password	The password for the current Predict file.	---
FDIC Cipher	The cipher for the current Predict file.	---
Adabas version	The version of Adabas for which the global format buffer was generated. Select a version from the list. If you want special fields and descriptors included in the generated GFB and corresponding field tables, select the version that includes the phrase "Collation DEs are in Record/Format Buffer".	
Occurrences used	How multiple occurrences of PE and MU fields are generated in the GFB and resulting field table. A value of "Use Maximum" indicates that the maximum number of occurrences should be generated (191); a value of "No" indicates that no occurrences will be generated; a value of "Use value defined in the dictionary" indicates that the number of occurrences defined by the Predict `Occ` attribute should be generated.	"Use value defined in the dictionary"

You cannot modify the following fields on this screen. These display-only fields are described in the following table:

Parameter Name	Displays
User	The user ID of the user who generated the global format buffer.
Date	The date the global format buffer was generated.
Time	The time of day the global format buffer was generated.
FDIC DBID	The current database of the Predict file.
FDIC FNR	The current file number of the Predict file.

Parameter Name	Displays
Full format	Whether the full format buffer was generated. The full format buffer includes the length and format of Adabas fields. A value of "Yes" indicates that the full format buffer was generated; a value of "No" indicates it was not.

3 When you have supplied values for the fields, click **OK** to start generating the global format buffer.

A **Global Format Buffer** table appears in detail-view showing the generated GFB.

The Type column identifies the field type. *SQL Significance Indicator* fields are marked with an "_S"; counter fields are marked with "CM" or "CP" in the Type column.

4 Optionally, modify the **Cs**, **Read Only**, and **Key Type** settings for the fields, as described in the following table:

Setting	Valid Values	Description
Cs	"C" or blank	This setting indicates whether or not the field content should be converted to character format in the destination user exit. The Cs setting is only modifiable for non-counter fields in binary format with lengths of 1-8 characters. Valid values are C (convert to the field content to character) or blank (do not convert the field content). **Note:** This field cannot be modified for counter fields (`Ty` field set to "CM" or "CP") and SQL Significance Indicator fields (`Ty` field set to "_S").
Read Only	"Y" or blank	This setting indicates whether or not the field is read-only. Read-only fields are not replicated. When this is checked, it indicates that the field is read-only (not replicated); a blank value indicates that the field is not read-only and is replicated. **Note:** When you initially generate or regenerate the GFB and field table, the value of the **Read Only** setting for all fields is blank.
Key Type	"Descriptor", "Primary Key", "UQ Descriptor", or "no key"	This setting indicates the kind of key field that the field represents. Possible values are: ■ "Descriptor" -- identifies a descriptor key field ■ "Primary Key" -- identifies a primary key field **Note:** At this time, multiple-value (MU) and periodic group (PE) fields cannot be specified as primary keys or in composite keys if Event Replicator Target Adapter processing will be invoked by the subscription (if the DCLASS parameter of the destination used by the subscription is set to "SAGTARG"). ■ "UQ Descriptor" -- identifies a UQ descriptor key field ■ "no key"-- identifies a non-key field

Setting	Valid Values	Description
		Note: When you initially generate or regenerate the GFB and field table, the key fields are set to "Descriptor", "UQ Descriptor", or "no key", depending on the Predict definitions for those fields.

5 When all specifications have been made to your satisfaction, click **OK** to save the GFB definition.

Importing a Data Mapping Tool GFB and Field Table

If you have generated a GFB and field table using the Data Mapping Tool, you can import it into Event Replicator Administration and save it on the Replicator system file. Once it is stored on the Replicator system file, it can be used with the Event Replicator Target Adapter to replicate data to an RDBMS. For complete information on using the Data Mapping Tool, read *Using the Data Mapping Tool*, in *Data Mapping Tool User's Guide*.

Data Mapping Tool-generated GFBs and field tables can be imported into new and existing GFB definitions. If you want to import them into an existing GFB definition, the existing GFB definition must not already have an associated field table defined. This section covers the following topics:

- Importing a Data Mapping Tool GFB and Field Table into a New GFB Definition
- Importing a Data Mapping Tool GFB and Field Table into an Existing GFB Definition

Importing a Data Mapping Tool GFB and Field Table into a New GFB Definition

To import a Data Mapping Tool-generated GFB and field table into a new GFB definition:

1 List the GFB definitions in Event Replicator Administration, as described in ***Listing GFB Definitions***, elsewhere in this guide.

The GFB definitions are listed in detail-view.

2 Right-click on **Global Format Buffers** in the tree-view under **Replication Definitions** and click on **Create New Format Buffer** in the resulting drop-down menu.

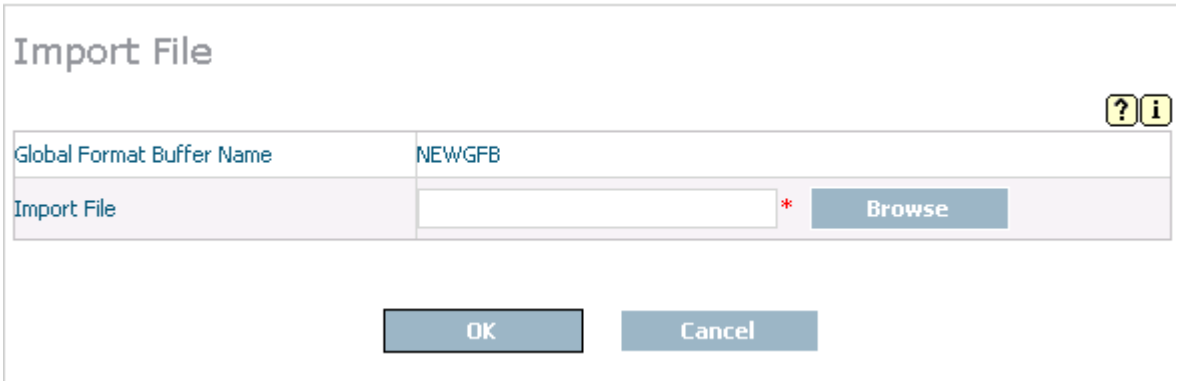
A drop-down menu appears.

3 Click on **Create New Format Buffer** in the drop-down menu.

A blank **Create New Global Format Buffer** panel appears in detail-view.

4 Specify a unique name for the GFB definition in the **Global Format Buffer Name** field and click the **Import** button.

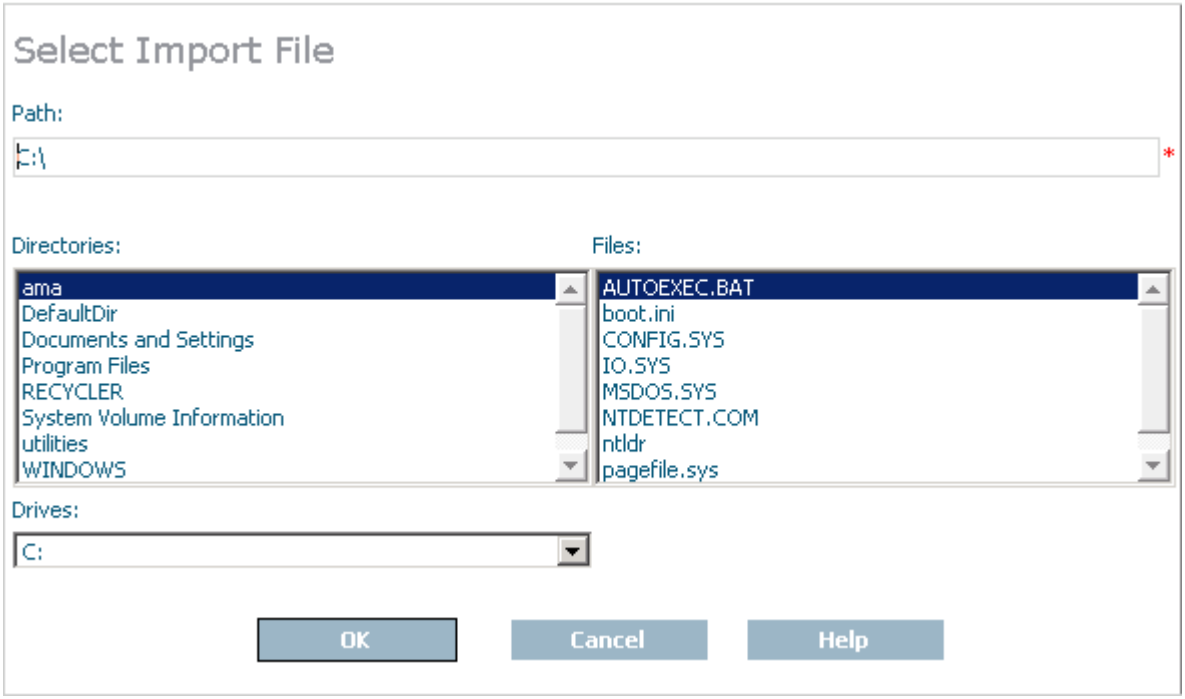
The **Import File** panel appears in detail-view.



5 In the **Import File** field on the **Import File** panel, specify the fully qualified path name of the file containing the Data Mapping Tool-generated GFB and field table.

Or:

Click on the **Browse** button to locate the file. The **Select Import File** panel appears allowing you to locate and select the file.



6 Use the drop-down boxes and other fields on the **Select Import File** panel to select the drive, directory, and file name of the file. When the file has been selected, click **OK** to return to the **Import File** panel.

The **Import File** panel appears with the name of the Data Mapping Tool-generated file specified in the **Import File** field.

7 Click **OK** on the **Import File** panel.

The Data Mapping Tool-generated file is imported into Event Replicator Administration, stored in the Replicator system file, and a **Global Format Buffer** panel appears in detail-view, showing the GFB.

Importing a Data Mapping Tool GFB and Field Table into an Existing GFB Definition

To import a Data Mapping Tool-generated GFB and field table into an existing GFB definition:

1 List the GFB definitions in Event Replicator Administration, as described in *Listing GFB Definitions*, elsewhere in this guide.

The GFB definitions are listed in detail-view.

2 Select an existing GFB definition in the GFB list

The existing GFB definition appears in detail-view.

3 Click on the **Modify** button.

The existing GFB definition still appears in detail-view, but now you have the option of generating a field table using Predict (read *Generating a GFB*, elsewhere in this section for more information) or of importing a GFB file with an associated field table.

4 Click the **Import** button.

The **Import File** panel appears in detail-view.

5 In the **Import File** field on the **Import File** panel, specify the fully qualified path name of the file containing the Data Mapping Tool-generated GFB and field table.

Or:

Click on the **Browse** button to locate the file. The **Select Import File** panel appears allowing you to locate and select the file. Use the drop-down boxes and other fields on this panel to select the drive, directory, and file name of the file. When the file has been selected, click **OK** to return to the **Import File** panel.

The **Import File** panel appears with the name of the Data Mapping Tool-generated file specified in the **Import File** field.

6 Click **OK** on the **Import File** panel.

The Data Mapping Tool-generated file is imported into Event Replicator Administration, stored in the Replicator system file, and a **Global Format Buffer** panel appears in detail-view, showing the GFB.

Reviewing GFB Generation Information

If you have **generated a GFB using Predict** or if you have **imported a GFB and field table that were generated using the Data Mapping Tool**, you can review the GFB generation information. This includes, but is not limited to statistics such as the file ID (table name) included in the GFB and the date and time the GFB was generated.

To review the generation information of a GFB:

1 List the GFB definitions in Event Replicator Administration, as described in *Listing GFB Definitions*, elsewhere in this guide.

The GFB definitions are listed in detail-view.

2 Select a generated or imported GFB from the list.

The GFB Definition panel appears in detail-view listing the current settings for the GFB definition you selected.

3 Click on the  icon in the upper left corner of the detail-view display.

The **Generation Information** is listed in detail-view.

Modifying GFB Definitions

To use Event Replicator Administration to modify a global format buffer (GFB) definition in the Replicator system file:

1 List the GFB definitions in Event Replicator Administration, as described in *Listing GFB Definitions*, elsewhere in this guide.

The GFB definitions are listed in detail-view.

2 Locate the definition you want to modify in the table in detail-view and click on it.

The GFB Definition panel appears in detail-view listing the current settings for the GFB definition you selected.

3 Click the **Modify** button.

The GFB parameters you can modify become editable in detail-view. For information on modifying these parameters, read the descriptions of *Adding GFB Definitions* and *Generating a GFB*, elsewhere in this section.

4 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the changes.

Copying GFB Definitions

To use Event Replicator Administration to copy a global format buffer (GFB) definition in the Replicator system file:

1 List the GFB definitions in Event Replicator Administration, as described in *Listing GFB Definitions*, elsewhere in this guide.

The GFB definitions are listed in detail-view.

2 Locate the definition you want to copy in the table in detail-view and click on it.

The GFB Definition panel appears in detail-view listing the current settings for the GFB definition you selected.

3 Click the **Copy** button.

A copy of the GFB definition is created and its parameter values appear in detail-view.

4 Specify a new, unique name for the copy of the GFB definition in the Value column for the **Global Format Buffer Name** parameter.

5 If you wish, modify any other parameters for the new copy in detail-view. For information on modifying the parameters, read the description of *Adding GFB Definitions*, elsewhere in this section.

6 When all modifications have been made, click **OK** to save the changes or click **Cancel** to cancel the copy.

Deleting GFB Definitions

To use Event Replicator Administration to delete a global format buffer (GFB) definition in the Replicator system file:

1 List the GFB definitions in Event Replicator Administration, as described in *Listing GFB Definitions*, elsewhere in this guide.

The GFB definitions are listed in detail-view.

2 Locate the definition you want to delete in the table in detail-view and click on it.

The GFB Definition panel appears in detail-view listing the current settings for the GFB definition you selected.

3 Click the **Delete** button.

A confirmation panel appears verifying that you want to delete the definition. If you click **Yes** (indicating that you do want to delete the definition), the definition is deleted. If you click **No** (indicating that you do not want to delete the definition), the definition is not deleted.

13 Reviewing Active Replication Parameters of Event Replicator Servers

If you are reviewing the active parameters of an Event Replicator Server, you have the option to review the active replication parameters of that database.

To display the replication parameter settings for an Event Replicator database:

1 Select an Event Replicator Server in tree-view as described in *Selecting Event Replicator Databases*, elsewhere in this guide.

2 Click and expand **Active Parameters** in tree-view under the database name.

The active ADARUN parameters for the database are displayed in tables in detail-view.

3 Click and expand **Replication** in tree-view under **Active Parameters**.

The general (global value) replication parameters for the Event Replicator database are listed in detail-view.

For explanations of each of these replication parameters, read *Setting Global Values*, elsewhere in this guide.

4 If you want to review the replication parameters for specific types of definitions, select the appropriate definition type in tree-view under **Replication**.

A table listing the definitions of the type you selected appears in detail-view. To see detail statistics for a given definition, click on it in the table.

For explanations of each of the replication parameters, read *Maintaining Replication Definitions* , elsewhere in this guide.

14 Reviewing the Replication Statistics of Event Replicator Databases

You can review the replication statistics of Event Replicator databases. These statistics are split into six types: general replication statistics, subscription statistics, destination statistics, input queue statistics, subtask statistics, and token statistics, as described in this chapter.

Reviewing General Replication Statistics

To review the general replication statistics of an Event Replicator database:

1 Select an Event Replicator Server in tree-view as described in *Selecting Databases*, elsewhere in this guide.

2 Click on **System Status** in tree-view under the Event Replicator Server name.

The system status information for the database is displayed in tables on the **System Status** panel in detail-view.

3 Click on **Replication Statistics** in tree-view under **System Status**.

The general replication statistics for the database appears on the **Replication Statistics Overview** panel in detail-view.

The general information includes values for the following properties:

Category	Property	Explanation
Replicated Transactions	Total Transactions	The total number of transactions that have been replicated by the Event Replicator Server.
	Pending	The number of transactions waiting to be replicated by the Event Replicator Server.
Replicated Messages	Total Messages	The total number of messages received by the Event Replicator Server.
Replicated Bytes	Number of Bytes sent to all Destinations	The total number of bytes of data sent to all destinations defined to the Event Replicator Server.
SLOG Statistics for Items	Delogged	The total number of items delogged by the Event Replicator Server.
	Logged	The total number items logged by the Event Replicator Server.
	On File	The total number of items on the SLOG of the Event Replicator Server.
Received from Input Queues	Messages	The total number of messages received from the input queues defined to the Event Replicator Server.
	Bytes	The total number of bytes received from the input queues defined to the Event Replicator Server.
	Commits	The total number of commits received from the input queues defined to the Event Replicator Server.

Category	Property	Explanation
	Backouts	The total number of backouts issued by the input queues defined to the Event Replicator Server.
Deleted Totals from Input Queues	Messages	The total number of messages deleted from Event Replicator Server input queues.
	Bytes	The total number of bytes deleted from Event Replicator Server input queues.
	Commits	The total number of commit requests deleted from Event Replicator Server input queues.
	Backouts	The total number of backout requests deleted from Event Replicator Server input queues.
Deleted Destinations Totals	Bytes	The total number of bytes deleted from Event Replicator Server destinations.
Database-related Input Transactions	Delogged	The total number of transactions delogged by the database.
	Logged	The total number of transactions logged by the database.
	On File	The total number of transactions on the SLOG of the database.
	With FCB on File	The total number of transactions with the FCB on file.

Detailed SLOG statistics are then listed by Event Replicator for Adabas destination definition. These statistics include the number of items processed from the SLOG to the destination, the number of items logged to the SLOG for the destination, the total number of items in the SLOG for the destination, and the number of items in the SLOG for the destination that are to be deleted.

In addition, detailed SLOG statistics are also given listed by Event Replicator for Adabas destination, input queue, subscription, subtask, and token definition.

Finally, replay processing (ADARPL) statistics are listed by database ID, if any are available.

Reviewing Destination Statistics

To review the destination statistics of an Event Replicator Server:

1 Select an Event Replicator Server in tree-view as described in *Selecting Databases*, elsewhere in this guide.

2 Click on **System Status** in tree-view under the Event Replicator Server name.

The system status information for the database is displayed in tables on the **System Status** panel in detail-view.

3 Click on **Replication Statistics** in tree-view under **System Status**.

The general replication statistics for the database appears on the **Replication Statistics Overview** panel in detail-view.

4 Click on **Destination** in tree-view under **Replication Statistics**.

A table appears in detail-view listing the total number of transactions replicated by each destination defined to this Event Replicator Server. The total number of pending transactions awaiting replication are also listed. The table title is **Destination Statistics**.

5 Click on a destination name in the **Destination Statistics** table.

Detailed statistics about the use of the selected destination appear in detail-view.

Reviewing Input Queue Statistics

To review the input queue statistics of an Event Replicator Server:

1 Select an Event Replicator Server in tree-view as described in *Selecting Databases*, elsewhere in this guide.

2 Click on **System Status** in tree-view under the Event Replicator Server name.

The system status information for the database is displayed in tables on the **System Status** panel in detail-view.

3 Click on **Replication Statistics** in tree-view under **System Status**.

The general replication statistics for the database appears on the **Replication Statistics Overview** panel in detail-view.

4 Click on **Input Queue** in tree-view under **Replication Statistics**.

A table appears in detail-view listing some general statistics about the input queue definitions defined to this Event Replicator Server. The table title is **Input Queue Statistics**.

5 Click on an input queue name in the **Input Queue Statistics** table.

Detailed statistics about the use of the selected input queue definition appear in detail-view.

Reviewing Subscription Statistics

To review the subscription statistics of an Event Replicator Server:

1 Select an Event Replicator Server in tree-view as described in *Selecting Databases*, elsewhere in this guide.

2 Click on **System Status** in tree-view under the Event Replicator Server name.

The system status information for the database is displayed in tables on the **System Status** panel in detail-view.

3 Click on **Replication Statistics** in tree-view under **System Status**.

The general replication statistics for the database appears on the **Replication Statistics Overview** panel in detail-view.

4 Click on **Subscription** in tree-view under **Replication Statistics**.

A table appears in detail-view listing the total number of transactions replicated by each subscription defined to this Event Replicator Server. The table title is **Subscription Statistics**.

5 Click on a subscription name in the **Subscription Statistics** table.

Detailed statistics about the use of the selected subscription appear in detail-view.

Reviewing Subtask Statistics

To review the subtask statistics of an Event Replicator Server:

1 Select an Event Replicator Server in tree-view as described in *Selecting Databases*, elsewhere in this guide.

2 Click on **System Status** in tree-view under the Event Replicator Server name.

The system status information for the database is displayed in tables on the **System Status** panel in detail-view.

3 Click on **Replication Statistics** in tree-view under **System Status**.

The general replication statistics for the database appears on the **Replication Statistics Overview** panel in detail-view.

4 Click on **Subtask** in tree-view under **Replication Statistics**.

A table appears in detail-view listing the total number of transactions replicated by each Event Replicator subtask. The table title is **Subtask Statistics**.

5 Click on a subtask name in the **Subtask Statistics** table.

Detailed statistics about the use of the selected subtask appear in detail-view.

Reviewing Token Statistics

To review the token statistics of an Event Replicator Server:

1 Select an Event Replicator Server in tree-view as described in *Selecting Databases*, elsewhere in this guide.

2 Click on **System Status** in tree-view under the Event Replicator Server name.

The system status information for the database is displayed in tables on the **System Status** panel in detail-view.

3 Click on **Replication Statistics** in tree-view under **System Status**.

The general replication statistics for the database appears on the **Replication Statistics Overview** panel in detail-view.

4 Click on **Token** in tree-view under **Replication Statistics**.

A table appears in detail-view listing token statistics. The table title is **Token Statistics**.

5 Click on a toekn name in the **Token Statistics** table.

Detailed statistics about the use of the selected token appear in detail-view.

15 Managing File Replication of Adabas Files

You can manage file replication for those Adabas databases for which replication has been turned on (the ADARUN REPLICATION parameter is set to YES).

Defining File Replication

You can define some of the replication properties of a file in an Adabas database for which replication has been turned on.

To define the file replication properties for an Adabas database file:

1 List the file properties for a file on an Adabas database for which replication has been turned on. For information on listing file properties, read *Reviewing File Properties*, in *Adabas Manager Documentation*.

2 At the bottom of the file properties screen, click on the **Define Replication** button.

Note: The **Define Replication** button is not available if replication is already defined for the file.

The **Define Replication** panel appears.

3 Modify the values for the editable properties on this panel, described in the following table. Click in the "Value" field for the property to modify the property.

Property Name	Description
Replication Target ID (RPLTARGETID)	Specify the DBID of the Event Replicator Server to which the Adabas file should to be replicated.
Replication Primary Key (RPLKEY)	Specify the primary key for replication
Replication of Before Images of Updates (RPLDSBI)	Indicate whether the collection of data storage before images during an update should occur. Click in the check box until a check mark appears if you want to collect data storage before images during an update.

4 When all properties have been specified as you want, click on the **OK** button to save the definitions or the **Cancel** button to cancel them.

Modifying File Replication Properties

You can modify some of the replication properties of a file in an Adabas database for which replication has been turned on.

To modify the file replication properties for an Adabas database file:

1 List the file properties for a file on an Adabas database for which replication has been turned on. For information on listing file properties, read *Reviewing File Properties*, in *Adabas Manager Documentation*.

2 At the bottom of the file properties screen, click on the **Modify Replication** button.

Note: The **Modify Replication** button is not available if replication has been removed for the file.

The **Modify Replication** panel appears.

3 Modify the value for one of the editable properties on this panel, described in the following table. Click in the Value field for the property to modify the property.

Property Name	Description
Replication Primary Key (RPLKEY)	Specify the primary key for replication
Replication of Before Images of Updates (RPLDSBI)	Indicate whether the collection of data storage before images during an update should occur. Click in the check box until a check mark appears if you want to collect data storage before images during an update.

4 When all properties have been specified as you want, click on the **OK** button to save the changes or the **Cancel** button to cancel them.

Activating File Replication

You can activate replication for a file in an Adabas database for which replication has been turned on.

To activate replication for an Adabas database file:

1 List the file properties for a file on an Adabas database for which replication has been turned on. For information on listing file properties, read *Reviewing File Properties*, in *Adabas Manager Documentation*.

2 At the bottom of the file properties screen, click on the **Activate Replication** button.

Note: The **Activate Replication** button is not available if replication is already activated for the file.

Replication is activated for the file.

Deactivating File Replication

You can deactivate replication for a file in an Adabas database for which replication has been turned on.

To deactivate replication for an Adabas database file:

1 List the file properties for a file on an Adabas database for which replication has been turned on. For information on listing file properties, read *Reviewing File Properties*, in *Adabas Manager Documentation*.

2 At the bottom of the file properties screen, click on the **Deactivate Replication** button.

Note: The **Deactivate Replication** button is not available if replication is already deactivated for the file.

Replication is deactivated for the file.

Removing File Replication

You can turn off replication and reset the replication properties of a file in an Adabas database for which replication has been turned on.

To turn off replication and reset the file replication properties for an Adabas database file:

1 List the file properties for a file on an Adabas database for which replication has been turned on. For information on listing file properties, read *Reviewing File Properties*, in *Adabas Manager Documentation*.

2 At the bottom of the file properties screen, click on the **Remove Replication** button.

Note: The **Remove Replication** button is not available if replication has already been removed for the file.

Replication is removed for the file.

16 Initiating a Replay Request Using Event Replicator Administration

This chapter describes how you can initiate synchronized and replay-only replay processing. This method involves a combination of Event Replicator Administration and a batch ADARPL utility job or automated replay. You first use Event Replicator Administration to generate a replay request. The replay request is assigned a token that you then use in the batch ADARPL utility job. For complete information about the ADARPL utility, read *ADARPL Utility: PLOG Replication Replay* in *Event Replicator for Adabas Reference Guide* provided with your Event Replicator for Adabas documentation.

To generate a synchronized or replay-only replay request using Event Replicator Administration and the ADARPL utility, complete the following steps:

1 Select an Event Replicator Server in tree-view as described in ***Selecting Event Replicator Databases***, elsewhere in this guide.

2 Click on **Replay** in the tree-view under **Replicators**.

A table listing the specifications you must make to initiate replay processing appears in detail-view.

3 Update the fields in detail-view as described in this table.

Parameter Name	Specify	Default
DBID	The database ID of the Adabas database from which you want replicated transactions replayed.	---
Automated	A check in this checkbox if you want the replay automated or not. An automated replay will automatically perform steps **11 through 13** of this procedure. A non-automated replay will not perform these steps automatically, and you will need to perform them manually. For complete	checked

Parameter Name	Specify	Default
	information about automating replay processing, read *Automating Replay Processing*, in the *Event Replicator for Adabas Administration and Operations Guide*. **Note:** If the `RECORDPLOGINFO` parameter has been set to NO, you cannot run an automated replay.	
Timeout	Optionally, specify the length of time, in seconds, at which the replay request should time out.	900 seconds
From Date	The date from which replicated transactions should be replayed. Dates should be specified in YYYY/MM/DD format. Replay processing will start with transactions in the PLOG that ended at or after this date. The combination of From Date and From Time must be earlier than the current date and time and earlier than the specified From Date and Time.	---
From Time	The time from which replicated transactions should be replayed. Times should be specified in HH:MM:SS format. Replay processing will start with transactions in the PLOG that ended at or after this time. The combination of From Date and From Time must be earlier than the current date and time and earlier than the specified From Date and Time.	---
To Date	The date to which replicated transactions should be replayed. Dates should be specified in YYYY/MM/DD format. Replay processing will stop with transactions in the PLOG that ended before this date. The combination of To Date and To Time must be later than the specified From Date and Time. If no end date is specified, the end date is the current date (the date the replay request is issued).	The current date
To Time	The time to which replicated transactions should be replayed. Times should be specified in HH:MM:SS format. Replay processing will stop with transactions in the PLOG that ended before this time. The combination of To Date and To Time must be later than the specified From Date and Time. If no end time is specified, the end time is the current time (the time the replay request is issued).	The current time

4 Click on the checkmark in the **Value** column of the **Destination Name List** field.

The destination list tables appear in detail-view.

5 Optionally, specify a list of destinations for replay processing in the **Destinations Selected** table. When the replay request is initiated, transactions will be replayed that were originally destined for the destinations on this list.

- To select a destination, click on a destination name in the **Destinations Available** table and then click the right arrow. The selected destination will appear in the **Destinations Selected** table. You can use the **Up** and **Down** buttons to order the destinations in the list.

- To remove a destination from the **Destinations Selected** table, select it and click the left arrow. The selected destination is moved back to the **Destinations Available** table.

6 When you are satisfied with the list of destinations in the **Destinations Selected** table, click **OK** to accept them and return to the Initiate Replication Replay screen.

Note: The destinations you selected appear in the **Destination Name List** field.

7 Click on the checkmark in the **Value** column of the **Subscription Name List** field.

The subscription list tables appear in detail-view.

8 Optionally, specify a list of subscriptions for replay processing on the **Subscriptions Selected** table. When the replay request is initiated, transactions will be replayed that were originally initiated by the subscriptions on this list.

- To select a subscription, click on a subscription name in the **Subscriptions Available** table and then click the right arrow. The selected subscription will appear in the **Subscriptions Selected** table. You can use the **Up** and **Down** buttons to order the subscriptions in the list.
- To remove a subscription from the **Subscriptions Selected** table, select it and click the left arrow. The selected subscription is moved back to the **Subscriptions Available** table.

9 When you are satisfied with the list of subscriptions in the **Subscriptions Selected** table, click **OK** to accept them and return to the Initiate Replication Replay screen.

Note: The subscriptions you selected appear in the **Subscription Name List** field.

10 When all fields on the Initiate Replication Replay screen have been filled in to your satisfaction, click **Synchronized** or **Replay Only** to initiate the appropriate replay request. Synchronized and Replay Only are two of the three possible replay modes in which ADARPL can run. For complete information about replay modes, read *Understanding Replay Modes* in the Event Replicator for Adabas Administration and Operations Guide.

The replay request is generated and a replay token is assigned to it. This replay token is displayed in an Event Replicator Administration message and in the Event Replicator Server job log.

Make a note of this token number as it is used in **step 13** if you are initiating replication replay using a batch ADARPL job.

If you have automated replication replay processing, this token number is picked up automatically by the generated replay jobstream and you can skip the remaining steps in this procedure. For complete information about automating replay processing, read *Automating Replay Processing*, in the *Event Replicator for Adabas Administration and Operations Guide*.

11 This step should not be performed if an automated replay is requested (**Automated** is checked on the Initiate Replication Replay screen).

If necessary, issue a force-end-of-PLOG request to the Adabas database and wait until the resulting PLCOPY job has copied or merged the latest PLOG data set. This is necessary only when the PLOG for the selected replay end date and time has not yet been copied or merged, for example, if no end date and time were specified in the replay request. This is also only necessary if an automated replay was not selected (**Automated** is not checked on the Initiate Replication Replay screen).

12 This step should not be performed if an automated replay is requested (**Automated** is checked on the Initiate Replication Replay screen).

Identify the sequential PLOG data sets that contain the protection data for the replicated records you need replayed. The PLOG data sets must build a complete sequence from the PLOG that includes the replay processing start time to the latest PLOG you copied or merged in the previous step.

13 This step should not be performed if an automated replay is requested (**Automated** is checked on the Initiate Replication Replay screen).

Run an ADARPL utility job, using the syntax described in *Syntax for Initiating ADARPL With A Token* in *Event Replicator for Adabas Reference Guide* provided with your Event Replicator Administration documentation. Be sure to specify:

- A concatenated list of the PLOG data sets you identified in the previous step.
- The replay request token assigned in **step 10**. This token should be specified in the ADARPL TOKEN parameter.
- The Event Replicator Server ID of the Event Replicator Server to which the replayed transactions should be sent. This token should be specified in the ADARPL RPLTARGETID parameter.

For more information about using the ADARPL utility, in general, read *ADARPL Utility: PLOG Replication Replay* in *Event Replicator for Adabas Reference Guide* provided with your Event Replicator Administration documentation.

The replay process is initiated using the replay request generated in Event Replicator Administration.

17 Reviewing and Managing the PLOG Data Set List

Event Replicator Administration replay processing provides a list of PLOG data sets for which replay processing can be initiated. This list also displays the date and time of the earliest transaction in each PLOG data set. You can use this list to review information about each PLOG and determine which PLOGs you need to replay. You can also delete PLOG data set entries from this list. This might be useful if the PLOG data set list becomes too long. For example, very old PLOG data sets appearing in the list may no longer be of use to you. In this case, you can use this screen to remove their PLOG entries from the list.

Note: The list of PLOG data sets may not appear or may contain inaccurate information, if the `RECORDPLOGINFO` initialization parameter is not set to "YES". You can control this parameter setting from Event Replicator Administration using the **Record PLOG Information** field on the global values screen. For more information about the `RECORDPLOGINFO` initialization parameter, read *RECORDPLOGINFO Parameter* in *Event Replicator for Adabas Reference Guide* provided with your Event Replicator Administration documentation. For more information about the **Record PLOG information** field, read ***Setting Global Values***, elsewhere in this guide.

Listing the PLOG Data Sets

To review the list of PLOG data sets for which replay processing can be initiated:

1 Select an Event Replicator Server in tree-view as described in ***Selecting Event Replicator Databases***, elsewhere in this guide.

2 Click on **PLOG Information** in the tree-view under **Replicators**.

A table listing the PLOG data sets appears in detail-view.

3 The following information is displayed on this screen:

Field	Description
Database ID	The ID of the database associated with the PLOG data set.
Date	The date of the first transaction in the PLOG data set.
PLOG Data Set Name	The name of a PLOG data set for which replay processing can be initiated.
Time	The time of the first transaction in the PLOG data set.

Reviewing PLOG Information

You can review more information about each PLOG data set. This information can help you determine whether you want to replay records from the data set and whether or not you want to remove the PLOG data set entry from the list of PLOG data sets in the Replicator system file.

To review more information about a PLOG data set:

1 Access the list of PLOG data sets from which replay processing can be requested, as described in *Listing the PLOG Data Sets*, earlier in this section.

2 Locate the PLOG data set for which you want more information, and click on the date associated with it.

The Replay - PLOG Information table appears in detail-view.

Screen Field	Description
Database ID	The ID of the database associated with the PLOG data set.
PLOG Data Set Name	The name of the PLOG data set.
Start Date and Time	The date and time of the first transaction in the PLOG data set.
End Date and Time	The date and time of the last transaction in the PLOG data set.
PLOG Session Number	The number of the PLOG session for this PLOG data set.
From Block Number	The starting block number in this PLOG data set.
To Block Number	The ending block number in this PLOG data set.
Last PLOG in Session	Indicates whether or not the PLOG data set is the last PLOG data set in the session. A value of "Y" in this field indicates that it is the last PLOG data set in the session; a value of "N" indicates that it is not the last PLOG data set in the session.

Deleting PLOG Data Set Entries From the List

You may find it useful to delete PLOG data set entries from the list of PLOG data sets -- especially if the list gets too long and the Replicator system file starts to fill up. For example, very old PLOG data sets appearing in the list may no longer be of use to you. In this case, you can use this screen to remove their entries from the list.

Notes:

1. When you delete a PLOG data set entry from the list, all earlier PLOG data set entries are also deleted.
2. When you delete a PLOG data set entry from the list, the PLOG data set still exists. Only its entry in the Replicator system file is deleted.

To delete PLOG data set entries from the list of PLOG data sets:

1 Access the list of PLOG data sets from which replay processing can be requested, as described in *Listing the PLOG Data Sets*, earlier in this section.

2 Locate the PLOG data set for which you want more information, and click on the date associated with it.

The Replay - PLOG Information table appears in detail-view.

3 Click the **Delete** button.

The PLOG data set entry and all earlier PLOG data set entries are removed from the list.

18 Submitting Event Replicator Target Adapter Requests

You can use Event Replicator Administration to submit requests to the Event Replicator Target Adapter when it is activated. Ordinarily, once the Event Replicator Target Adapter processing is activated, replicated transaction data for the subscriptions and destinations are transformed and transferred to the Event Replicator Target Adapter to be applied to your RDBMS. In addition, though, you can request that the Event Replicator Target Adapter:

- Initiate a request for initial-state data for a specific subscription, database, Adabas file, and data values to populate the RDBMS tables.
- Clear the data in an RDBMS table, based on a specific subscription, database, and Adabas file.
- Delete the RDBMS tables for a specific subscription, database, and Adabas file.

Notes:

1. None of these requests can be processed if the Event Replicator Target Adapter is not fully activated. For complete information on fully activating Event Replicator Target Adapter processing, read *Activating Event Replicator Target Adapter Processing*, in *Event Replicator Target Adapter User Guide* provided with your Event Replicator Administration documentation.
2. Be sure you have the proper authorization privileges to maintain the RDBMS tables. Effective with Event Replicator Target Adapter 2.7, user authorization to maintain any new RDBMS tables via Event Replicator Target Adapter is inherited from the site's privilege settings for the database. Authorization is managed by the user's RDBMS privileges and not by Event Replicator Target Adapter. Event Replicator Target Adapter will no longer grant RDBMS privileges to the user. Therefore, if you want to use Event Replicator Target Adapter to maintain tables in an RDBMS, verify that your RDBMS authorization privileges are correct for the maintenance you want to perform.

This chapter describes the methods by which you can control Event Replicator Target Adapter behavior.

Populating the RDBMS Tables

Using Event Replicator Administration, you can request that the Event Replicator Target Adapter initiate a request to populate the RDBMS tables with initial-state data for specific subscriptions, databases, Adabas files, and data values. When you submit the RDBMS population request from Event Replicator Administration, it is sent to the Event Replicator Target Adapter, which receives it and submits its own initial-state request to the Event Replicator for the data. The data is then transferred to the RDBMS via the usual processing of the Event Replicator Target Adapter.

Note: If you submit a request to the Event Replicator Target Adapter to populate an RDBMS table that does not exist, Event Replicator Target Adapter will create the table in the course of its normal processing. However, if you submit a request to the Event Replicator Target Adapter to populate an existing RDBMS table, it is your responsibility to clear or drop the

existing table prior to populating the RDBMS table with new data. For information on clearing an RDBMS table, read ***Clearing (Refreshing) the RDBMS Table Data***, elsewhere in this guide; for information on dropping an RDBMS table, read ***Deleting (Dropping) RDBMS Tables***, elsewhere in this guide.

To submit a request to the Event Replicator Target Adapter to populate the RDBMS:

The Event Replicator Target Adapter must be installed and started or the request will not be processed until it is. For more information on installing, configuring, and starting the Event Replicator Target Adapter, read *Installing the Event Replicator Target Adapter*, *Event Replicator Target Adapter Administration*, and *Starting the Event Replicator Target Adapter*, in the Event Replicator Target Adapter documentation.

1 Select an Event Replicator Server in tree-view as described in ***Selecting Event Replicator Databases***, elsewhere in this guide.

2 Click and expand **Target Adapter** in the tree-view under **Replicators**.

A table listing the Event Replicator Target Adapter functions appears in detail-view.

3 Click on **Populate Table** in the tree-view under **Target Adapter**.

A table listing the fields you need to specify to populate the RDBMS appears in detail-view.

4 Supply values for the fields in detail view, as described in the following table:

Field Name	Required?	Description
Initial-State Name	Yes	From the drop-down list, select the name of the initial-state definition in the Replicator system file that you want used to populate the RDBMS.
Initial-State File	Yes	From the drop-down list, select the Adabas database DBID and file number in the database that you want used to populate the RDBMS.
Subscription Name	Yes	From the drop-down list, select the name of the subscription definition in the Replicator system file that you want used to populate the RDBMS.
Value Buffer	No	This field is not available for use at this time.

5 When values for all required fields have been supplied, click **OK**.

The request is submitted to the Event Replicator Target Adapter.

Clearing (Refreshing) the RDBMS Table Data

Using Event Replicator Administration, you can request that the Event Replicator Target Adapter initiate a request to clear the data in the RDBMS tables for specific subscriptions, databases, and Adabas files. When you submit the RDBMS refresh request from Event Replicator Administration, it is sent to the Event Replicator Target Adapter, which receives it and clears the data out of the related tables, leaving the tables themselves in place.

To submit a request to the Event Replicator Target Adapter to clear data in the RDBMS:

The Event Replicator Target Adapter must be installed and started or the request will not be processed until it is. For more information on installing, configuring, and starting the Event Replicator Target Adapter, read *Installing the Event Replicator Target Adapter*, *Event Replicator Target Adapter Administration*, and *Starting the Event Replicator Target Adapter*, in the Event Replicator Target Adapter documentation.

1 Select an Event Replicator Server in tree-view as described in ***Selecting Event Replicator Databases***, elsewhere in this guide.

2 Click and expand **Target Adapter** in the tree-view under **Replicators**.

A table listing the Event Replicator Target Adapter functions appears in detail-view.

3 Click on **Refresh/Drop Table** in the tree-view under **Target Adapter**.

A table listing the fields you need to specify to refresh or drop the RDBMS data appears in detail-view.

4 Supply values for the fields on in detail-view, as described in the following table:

Field Name	Required?	Description
DBID	Yes	From the drop-down list, select the DBID of the Adabas database associated with the RDBMS data you want to cleared.
File	Yes	From the drop-down list, select the number of the file in the Adabas database of the file associated with the RDBMS data you want to cleared.
Subscription Name	Yes	From the drop-down list, select the name of the subscription definition in the Replicator system file associated with the RDBMS data you want cleared.

5 When values for all required fields have been supplied, click **Refresh**.

The request is submitted to the Event Replicator Target Adapter.

Deleting (Dropping) RDBMS Tables

Using Event Replicator Administration, you can request that the Event Replicator Target Adapter delete the RDBMS tables (and their associated data) for specific subscriptions, databases, and Adabas files. When you submit the RDBMS refresh request from Event Replicator Administration, it is sent to the Event Replicator Target Adapter, which receives it and processes it.

To submit a request to the Event Replicator Target Adapter to delete tables in the RDBMS:

The Event Replicator Target Adapter must be installed and started or the request will not be processed until it is. For more information on installing, configuring, and starting the Event Replicator Target Adapter, read *Installing the Event Replicator Target Adapter*, *Event Replicator Target Adapter Administration*, and *Starting the Event Replicator Target Adapter*, in the Event Replicator Target Adapter documentation.

1 Select an Event Replicator Server in tree-view as described in ***Selecting Event Replicator Servers***, elsewhere in this guide.

2 Click and expand **Target Adapter** in the tree-view under **Replicators**.

A table listing the Event Replicator Target Adapter functions appears in detail-view.

3 Click on **Refresh/Drop Table** in the tree-view under **Target Adapter**.

A table listing the fields you need to specify to refresh or drop the RDBMS data appears in detail-view.

4 Supply values for the fields on this screen, as described in the following table:

Field Name	Required?	Description
DBID	Yes	From the drop-down list, select the DBID of the Adabas database associated with the RDBMS tables you want to delete.
File	Yes	From the drop-down list, select the number of the file in the Adabas database of the file associated with the RDBMS tables you want to delete.
Subscription	Yes	From the drop-down list, select the name of the subscription definition in the Replicator system file associated with the RDBMS tables you want to delete.

5 When values for all required fields have been supplied, click **Drop**.

The request is submitted to the Event Replicator Target Adapter for processing.

Index

E

F

G

I

K

L

M

N

O

P

Q

R

S

T

V

W